

02997.001739

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



Here Application of:

Laurent FROUIN, et al.

Application No.: 10/055,939

Filed: January 28, 2002

For: DEVICE AND METHOD FOR
TRANSMISSION IN A SWITCH

)
: Examiner: Unassigned
)
: Group Art Unit: 2151
)
:
:
) May 1, 2002
:

Commissioner for Patents
Washington, D.C. 20231

SUBMISSION OF PRIORITY DOCUMENT

Sir:

In support of Applicants' claim for priority under 35 U.S.C. § 119, enclosed
is a certified copy of the following French patent application:

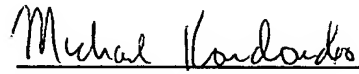
01 02037, filed February 14, 2001.



This Page Blank (uspto)

Applicants' undersigned attorney may be reached in our Washington, D.C. office by telephone at (202) 530-1010. All correspondence should continue to be directed to our below-listed address.

Respectfully submitted,



Attorney for Applicants
Michael E. Kondoudis
Registration No. 42,758

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

MEK/tmc

DC_MAIN 89666 v 1

This Page Blank (uspto)



CERTIFIED COPY OF
PRIORITY DOCUMENT



BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 30 JAN. 2002

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT
NATIONAL DE

SIEGE
26 bis, rue de Saint Petersburg
75800 PARIS cedex 08
Téléphone : 33 (1) 53 04 53 04

) ✓

This Data Blank (uspto)



BREVET D'INVENTION

26bis, rue de Saint-Pétersbourg
75800 Paris Cédex 08
Téléphone: 01 53.04.53.04 Télécopie: 01.42.94.86.54

Code de la propriété intellectuelle-livre VI

REQUÊTE EN DÉLIVRANCE

DATE DE REMISE DES PIÈCES N° D'ENREGISTREMENT NATIONAL DÉPARTEMENT DE DÉPÔT DATE DE DÉPÔT <i>n° 0102037</i> 14 FEV. 2001 14 FEV. 2001	Patrice VIDON LE NOBEL Technopole Atalante 2 Allée Antoine Becquerel 90333 35703 RENNES France
Vos références pour ce dossier: 6751	

1 NATURE DE LA DEMANDE			
Demande de brevet			
2 TITRE DE L'INVENTION			
		Dispositif et procédé de transmission dans un commutateur	
3 DECLARATION DE PRIORITE OU REQUETE DU BENEFICE DE LA DATE DE DEPOT D'UNE DEMANDE ANTERIEURE FRANCAISE		Pays ou organisation	Date N°
4-1 DEMANDEUR			
Nom	CANON KABUSHIKI KAISHA		
Rue	30-2, Shimomaruko 3-chome Ohta-Ku TOKYO JAPON		
Code postal et ville	Japon		
Pays	Japon		
Nationalité	KABUSHIKI KAISHA		
Forme juridique			
5A MANDATAIRE			
Identifiant	741A		
Nom	VIDON		
Prénom	Patrice		
Qualité	CPI: 92-1250		
Cabinet ou Société	CABINET VIDON		
Rue	LE NOBEL Technopole Atalante 2 Allée Antoine Becquerel 90333		
Code postal et ville	35703 RENNES		
N° de téléphone	02 99 38 23 00		
N° de télécopie	02 99 36 02 00		
6 DOCUMENTS ET FICHIERS JOINTS		Fichier électronique	Pages Détails
Désignation d'inventeurs			

Description	6751.doc	49		
Revendications	6751.doc	9	38	
Dessins		10	10 fig., 3 ex.	
Abrégé	6751.doc	1		
Listage de séquences				
Rapport de recherche				
Chèque		1 doc.	5392545	
7 RAPPORT DE RECHERCHE				
Etablissement immédiat				
8 REDEVANCES JOINTES	Devise	Taux	Quantité	Montant à payer
062 Dépôt	FRF	250.00	1.00	250.00
063 Rapport de recherche (R.R.)	FRF	2 100.00	1.00	2 100.00
068 Revendication à partir de la 11ème	FRF	115.00	28.00	3 220.00
Total à acquitter	FRF			5 570.00
9 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE				

La loi n°78-17 du 6 janvier 1978 relative à l'informatique aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

Dispositif et procédé de transmission dans un commutateur.

La présente invention se rapporte au domaine des réseaux de commutation, et en particulier à la transmission de données dans un réseau de commutation à haut débit.

On rappelle que la commutation permet d'aiguiller correctement des données entre une source et un ou plusieurs destinataires, ces données étant organisées en unités de données de protocole, et par exemple par paquets.

On connaît, dans l'art antérieur, deux techniques différentes de commutation (en anglais « switch »):

- la commutation de circuit ; et
- la commutation de paquets.

Dans un réseau à commutation de circuit, interconnectant plusieurs appareils, on utilise essentiellement un multiplexage par répartition dans le temps ou MRT (de l'anglais TDM ou « Time Division Multiplex »). Cette technique permet la transmission de données synchrones, par l'allocation d'intervalles de temps (en anglais « time slot ») sur le bus MRT pour des communications variées et synchronisées. En entrée, un appareil source écrit des données sur un ou plusieurs intervalles de temps qui lui sont alloués. En sortie, un ou plusieurs appareils cibles lisent les données sur l'intervalle de temps auxquelles elles sont attachées.

Selon cette technique, il est possible d'utiliser les intervalles de temps comme canal de diffusion, ceux-ci étant utilisés en écriture par une seule source et utilisés en lecture par plusieurs cibles. En outre, la nature synchrone du mécanisme garantit que toutes les données utiles sont traitées convenablement par tous les appareils, à la fois source et cible.

Dans un réseau à commutation de paquets, le transfert des données entre les appareils sources et les appareils cibles suppose que le réseau les reliant soit disponible. En d'autres termes, cela dépend du statut du réseau. Ainsi, on prévoit un protocole de type « poignée de main » connu sous le nom anglais de

« handcheck » afin d'assurer non seulement la disponibilité des données mais aussi leur bonne prise en compte.

Cela est particulièrement nécessaire, lorsqu'il s'agit de mettre en œuvre un « routage de type ver » (ou « wormhole routing » en anglais) dans lequel les données en cours de transmission peuvent être réparties entre plusieurs éléments intermédiaires des moyens de commutation (la décision de routage de type « wormhole routing » est prise avant la réception complète du paquet à l'opposé d'un « routage stockage et de relais » ou en anglais « store and forward routing ») et/ou lorsque des appareils sources et cibles ont des mémoires tampons de type FIFO (de l'anglais « First In First Out ») connectés à la fois à des ports sources et cibles.

Cela est encore plus critique dans certains cas, notamment lorsque un port cible doit faire face à de la congestion, ayant ainsi un débit relativement plus faible à la sortie d'une FIFO.

Par ailleurs, la technique à commutation de paquets ne permet pas d'envisager efficacement des émissions de type point vers multipoint (en anglais « multicast ») (c'est-à-dire lorsqu'un port source transmet des données vers au moins deux ports cibles), sauf à mettre en œuvre des coûteuses et complexes matrices de commutation dites « crossbar ».

Le rapport de thèse « Conception and realisation of high performance packet router », Reube, 3 juillet 1997 décrit un commutateur de type « ver » (8x8) reposant sur une implémentation de type « full crossbar » entre des ports parallèles.

La capacité de multipoint est garantie à travers des étiquetages d'intervalles, comme mis en œuvre par le composant C104 décrit dans le document de brevet US5,442,881 de Inmos Limited et le document de brevet US 5,088,091 de DEC qui illustre du routage de la diffusion à travers une table de routage. Les deux solutions reposent sur une implémentation de type commutateur « crossbar ».

Il n'existe pas aujourd'hui, selon l'état de l'art, de bus d'architectures à commutation dans des réseaux à commutation de paquets pour fournir des fonctions d'émission point vers multipoint, simples et efficaces. Les solutions basées sur des « crossbars » existent, mais restent complexes et coûteuses.

5 Un premier inconvénient de l'art antérieur est qu'il ne permet pas d'effectuer des commutations efficaces en toutes circonstances, en optimisant les performances (par exemple, le débit pour chaque communication ou le débit global).

Notamment, il n'existe pas de techniques sans commutation « crossbar »
10 qui s'avèrent efficaces et relativement peu coûteuses à mettre en œuvre :

- dans les cas où les commutations s'appliquent à des transmissions de type point à point ; et
- dans les cas où les commutations s'appliquent à des transmissions de type point-multipoint efficacement .

15 Ainsi, les systèmes connus favorisent généralement la commutation point à point.

Un autre inconvénient est que les techniques connues utilisant l'architecture de type « store and forward routing » présente une latence importante et nécessite une capacité mémoire plus importante.

L'invention selon ses différents aspects a notamment pour objectif de
20 pallier ces inconvénients de l'art antérieur.

Plus précisément, l'invention a pour objectif la fourniture d'une technique permettant une commutation point-multipoint, tout en permettant un débit optimal dans le cas des transmissions point à point.

En outre, l'invention a pour objectif de permettre de faire du routage type
25 « wormhole routing » en minimisant la latence de transfert de paquets à la fois pour les transmissions point à point et point à multipoint.

Encore un autre objectif est de fournir une architecture garantissant ainsi une implémentation à bas coût, et relativement peu complexe à mettre en œuvre.

Ces objectifs ainsi que d'autres qui apparaîtront par la suite sont atteints
30 selon l'invention, à l'aide d'un procédé de commutation d'unités de données de

protocole mettant en œuvre des moyens de commutation permettant d'interconnecter sélectivement un port émetteur et au moins un port récepteur sélectionné parmi au moins deux ports récepteurs, par l'intermédiaire d'au moins un bus interne, chacune des unités de données de protocole étant constituée d'au moins une donnée élémentaire, remarquable en ce qu'il met en œuvre :

- un mécanisme de synchronisation définissant des intervalles temporels, dits cycles de connexion, sur au moins un des bus internes ;
- un mécanisme d'allocation d'au moins un des cycles de connexion à chacun des ports récepteurs sélectionnés; et
- un mécanisme d'écriture d'au moins une donnée élémentaire dans le ou les cycles de connexion alloués, de façon à permettre la diffusion des données élémentaires aux ports récepteurs sélectionnés.

Une unité de données de protocole (ou « Protocol Data Unit » en anglais ou PDU en forme abrégée) est une unité de données échangées au niveau d'un protocole. Il peut s'agir par exemple d'un paquet comprenant au moins une donnée élémentaire.

Un cycle de connexion ou cycle de transmission d'unité de données de protocole est ici un cycle de transmission d'une unité de données complète. Ce cycle est généralement marqué par un signal particulier (par exemple le signal *Rx1-connected* utilisé dans le mode de réalisation préféré décrit plus loin) qui est validé durant toute la durée du cycle.

Une donnée élémentaire est typiquement une donnée binaire (ou « bit » en anglais) ou un multiplet (ou « byte » en anglais) de notamment 16, 32 ou 64 données binaires suivant la largeur des bus permettant le transfert des données dans un dispositif mettant en œuvre le procédé.

Ainsi, l'invention permet la mise en œuvre de transmissions point à point et point à multipoint effectuées selon un procédé de commutation synchronisé.

En outre, les performances sont optimisées dans toutes les conditions, notamment point à multipoint, diffusion et point à point.

Selon une caractéristique particulière, le mécanisme d'écriture comprend une étape de vérification déterminant si chacune des données élémentaires a été reçue par chacun des ports récepteurs sélectionnés.

5 Selon une caractéristique particulière, le mécanisme d'écriture de donnée élémentaire dans chacun des cycles de connexion alloués est réitéré tant que la donnée élémentaire n'a pas été reçue par tous les ports récepteurs.

Ainsi, toutes les données élémentaires d'une unité de données de protocole sont avantageusement reçues correctement par chacun des ports récepteurs destinataires de l'unité de données de protocole.

10 Selon une caractéristique particulière, le procédé est remarquable en ce que le mécanisme d'allocation comprend une étape d'association de chacun des cycles de connexion à chacun des ports récepteurs sélectionnés.

Ainsi, l'invention est avantageusement bien adaptée à traiter la commutation d'unités de données de protocole en mode point à multipoints.

15 Selon une caractéristique particulière, le mécanisme d'allocation comprend:

- une étape de détection d'un port émetteur requérant le transfert d'au moins une unité de données de protocole vers au moins un port récepteur sélectionné;
- une étape de vérification que le ou les ports récepteurs sélectionnés sont prêts
- 20 à recevoir la ou les unités de données de protocole ; et
- une étape de validation d'au moins un cycle de connexion permettant l'écriture des données élémentaires de la ou des unités de données de protocole dans les ports récepteurs sélectionné durant le ou les cycles de connexion validés, lorsque la vérification est positive.

25 Ainsi, de manière avantageuse, l'invention permet d'affecter au moins un cycle de connexion lorsque les ports récepteurs peuvent effectivement recevoir la ou les unités de données de protocole que ce soit en mode point à point ou point à multipoint.

30 Selon une caractéristique particulière, le procédé est remarquable en ce qu'il met en œuvre au moins un premier bus d'entrée multiplexant les données

élémentaires provenant d'au moins deux ports émetteurs et/ou au moins un premier bus de sortie multiplexant les données élémentaires à destination d'au moins deux ports récepteurs.

5 Selon une caractéristique particulière, les ports émetteurs et récepteurs sont organisés par paires regroupant chacune un port émetteur et un port récepteur, chaque paire étant associée à un lien distinct.

Selon une caractéristique particulière, le procédé est remarquable en ce qu'il met en œuvre au moins un lien connecté à des ports émetteurs et/ou à des ports récepteurs.

10 Selon une caractéristique particulière, le lien appartient au groupe comprenant :

- les liens IEEE 1355 ou équivalent; et
- les bus externes.

15 Selon une caractéristique particulière, le procédé est apte à commuter des données dont le débit est supérieur ou égal à 100 Mbit/s.

Ainsi, l'invention permet avantageusement la commutation de données à haut débit.

20 Selon une caractéristique particulière, le mécanisme d'écriture comprend au moins une étape d'écriture de chaque donnée élémentaire de chacune des unités de données de protocole, chacune des étapes d'écriture de chaque donnée élémentaire comprenant :

- une sous-étape d'acceptation par chacun des ports récepteurs sélectionné de l'écriture de chaque donnée élémentaire à transmettre ;
- et
- 25 - une sous-étape de transmission de chaque donnée élémentaire à transmettre, vers chacun des ports récepteurs sélectionné.

Ainsi, de façon préférentielle, l'invention permet la transmission d'un ou de plusieurs unités de données de protocole vers une pluralité de ports récepteurs, en gérant une transmission autorisée par chacun des ports récepteurs au niveau de

chaque donnée élémentaire contenue dans le ou les unités de données de protocole à transmettre.

En outre, l'invention permet avantageusement d'éviter les problèmes de congestion au sein des ports récepteurs tout en optimisant le débit global de données.

Selon une caractéristique particulière, lors de l'étape d'acceptation, l'acceptation est conditionnée par un degré de remplissage d'une mémoire de réception associée au port récepteur sélectionné, pour chacun des ports récepteurs.

Selon une caractéristique particulière, la mémoire comprend au moins une FIFO.

Selon une caractéristique particulière, le procédé est remarquable en ce que le mécanisme d'écriture comprend au moins une étape d'arbitrage d'au moins un bus reliant un ensemble d'au moins un port d'entrée comprenant le port émetteur à un ensemble d'au moins un port de sortie comprenant les ports récepteurs, l'étape d'arbitrage étant assurée par une matrice de commutation constituée de points de couplage apte à transmettre des données élémentaires entre un port d'entrée et un port de sortie, et organisée en lignes et en colonnes,

- chaque colonne (respectivement ligne) étant apte à gérer la réception de données élémentaires en provenance d'un port d'entrée associé à la colonne (respectivement ligne); et
- chaque ligne (respectivement colonne) étant apte à gérer l'émission de données élémentaires à destination d'un port de sortie associé à la ligne respectivement colonne);

de sorte qu'un seul point de couplage par ligne (respectivement colonne) puisse, à un instant donné, permettre la transmission de données élémentaires.

Ainsi, l'invention permet avantageusement une mise en œuvre simple et flexible du procédé qui permet d'optimiser la transmission de données notamment en point à multipoint (par duplication des données sur plusieurs lignes de la matrice de commutation).

On note que la matrice de commutation peut être :

- pleine (cas où des données élémentaires peuvent être reçues en provenance d'un port d'entrée quelconque et être émises vers un port de sortie quelconque ; en d'autres termes, toutes les commutations sont autorisées) ; ou
- 5 - creuse (cas où des données élémentaires en provenance de certains ports d'entrée ne peuvent être émises vers certains ports de sortie par construction de la matrice de commutation ; en d'autres termes, seules certaines commutations sont autorisées).

En outre, l'invention permet de manière préférentielle, une optimisation de
10 la synchronisation entre les opérations de lecture et d'écritures de données.

On note que le bus ou les bus peuvent être notamment de type MRT (c'est-à-dire à multiplexage par répartition dans le temps).

De plus, selon un mode de réalisation avantageux, l'invention permet l'établissement de plusieurs connexions simultanées, chaque ligne
15 (respectivement colonne) pouvant gérer une connexion.

L'invention permet aussi un chaînage des modules de commutation.

Selon une caractéristique particulière, le procédé est remarquable en ce que chaque unité de données de protocole transmise comprend au moins un en-tête et en ce que le procédé comprend en outre :

- 20 - au moins une étape d'analyse de l'en-tête ; et/ou
- au moins une étape de modification de l'en-tête.

Ainsi, l'invention permet avantageusement d'identifier les destinataires d'une unité de données de protocole transmise au cours de l'étape d'analyse.

L'invention concerne également un dispositif de commutation d'unités de
25 données de protocole mettant en œuvre des moyens de commutation permettant d'interconnecter sélectivement un port émetteur et au moins un port récepteur sélectionné parmi au moins deux ports récepteurs, par l'intermédiaire d'au moins un bus interne, chacune des unités de données de protocole étant constituée d'au moins une donnée élémentaire, remarquable en ce qu'il met en œuvre :

- un moyen de synchronisation définissant des intervalles temporels, dits cycles de connexion, sur au moins un des bus internes ;
- un moyen d'allocation d'au moins un des cycles de connexion à chacun des ports récepteurs sélectionnés; et
- 5 - un moyen d'écriture d'au moins une donnée élémentaire dans le ou les cycles de connexion alloués, de façon à permettre la diffusion des données élémentaires aux ports récepteurs sélectionnés.

Selon des caractéristiques particulières, le dispositif de commutation met
avantageusement en œuvre certains aspects du procédé de commutation décrit ci-
10 dessus.

Selon une caractéristique particulière, le dispositif est remarquable en ce qu'il comprend en outre un moyen d'interfaçage, délivrant à un module de contrôle et via un moyen de transmission de signaux d'horloges, des signaux d'horloge régénérés à partir de paquets reçus par le moyen d'interfaçage.

15 Selon une caractéristique particulière, le dispositif est remarquable en ce qu'il comprend un moyen d'interfaçage transmettant et/ou recevant des informations via au plus deux bus de liaison à destination et/ou en provenance d'au moins un des moyens appartenant au groupe comprenant le moyen de synchronisation, le moyen d'allocation et le moyen d'écriture.

20 Ainsi, le nombre de bus de liaison entre le moyen d'interfaçage et le ou les autres moyens est avantageusement limité, ce qui permet de simplifier la mise en œuvre du dispositif et/ou de réduire son coût.

25 Selon une caractéristique particulière, le dispositif est remarquable en ce que les unités de données de protocole émises par au moins un port émetteur vers des FIFOs sont multiplexées sur un bus de liaison en réception.

30 Selon une caractéristique particulière, le dispositif est remarquable en ce que les unités de données de protocole reçues par au moins un port récepteur par l'intermédiaire de FIFOs sont démultiplexées sur un bus de liaison en émission.

L'invention concerne, en outre, un appareil de commutation remarquable
en ce qu'il comprend :



- au moins le dispositif de commutation;
- et au moins un élément appartenant au groupe comprenant
 - les liens IEEE 1355 ou équivalent; et
 - les bus externes.

5 Selon une caractéristique particulière, l'appareil de commutation est relié à un appareil de traitement de données.

Ainsi, l'invention permet avantageusement la commutation de données à destination et/ou en provenance d'un appareil de traitement de données qui est, par exemple, un ordinateur, une imprimante, un appareil multimédia, un
10 lecteur/enregistreur audio et/ou vidéo, un appareil photographique, un appareil de visualisation d'images (écran, télévision...).

L'invention concerne également une application du procédé à l'un au moins des domaines appartenant au groupe comprenant :

- la commutation à haut débit ;
- 15 - les applications distribuées ;
- la transmission de données numériques ;
- la réception de données numériques ;
- les applications audio ;
- les réseaux d'entreprise ; et
- 20 - la transmission d'images en temps réel.

L'invention concerne aussi un produit programme d'ordinateur remarquable en ce que le programme comprend des séquences d'instructions adaptées à la mise en œuvre du procédé de commutation lorsque le programme est exécuté sur un ordinateur.

25 L'invention concerne, en outre, un produit programme d'ordinateur de commutation d'unités de données de protocole mettant en œuvre des moyens de commutation permettant d'interconnecter sélectivement un port émetteur et au moins un port récepteur sélectionné parmi au moins deux ports récepteurs, par l'intermédiaire d'au moins un bus interne, chacune des unités de données de
30 protocole étant constituée d'au moins une donnée élémentaire, le produit

programme d'ordinateur comprenant des instructions de code de programme enregistré sur un support utilisable dans un ordinateur comprenant :

- 5 - des moyens de programmation lisibles par ordinateur pour effectuer une étape de synchronisation définissant des intervalles temporels, dits cycles de connexion, sur au moins un des bus internes ;
- des moyens de programmation lisibles par ordinateur pour effectuer une étape d'allocation d'au moins un des cycles de connexion à chacun des ports récepteurs sélectionnés et
- 10 - des moyens de programmation lisibles par ordinateur pour effectuer une étape d'écriture d'au moins une donnée élémentaire dans le ou les cycles de connexion alloués, de façon à permettre la diffusion des données élémentaires aux ports récepteurs sélectionnés.

Les avantages des dispositif de commutation, appareil, application et produit programme d'ordinateur sont les mêmes que ceux du procédé de
15 commutation, ils ne sont pas détaillés plus amplement.

D'autres caractéristiques et avantages de l'invention apparaîtront plus clairement à la lecture de la description suivante d'un mode de réalisation préférentiel, donné à titre de simple exemple illustratif et non limitatif, et des dessins annexés, parmi lesquels :

- 20 - la figure 1 présente un synoptique de commutateur conforme à l'invention selon un mode particulier de réalisation ;
- la figure 2 présente un synoptique détaillé d'un un moyen de contrôle de flux de données, d'une interface avec un lien série appartenant au commutateur illustré en figure 1 ainsi que leurs interconnexions;
- 25 - la figure 3 représente un schéma électronique du module de commutation de la figure 2 ;
- la figure 4 présente un schéma électronique d'un moyen d'arbitrage (ou matrice de commutation) présent dans le module des figures 2 et 3 ;
- la figure 5 décrit un point de commutation de la matrice de
30 commutation de la figure 4 ;



- la figure 6 présente un point central de commutation régissant une colonne de la matrice de commutation de la figure 4 ;
- la figure 7 illustre un algorithme d'arbitrage dans un point de couplage présent dans la matrice de commutation de la figure 4 ;
- 5 - la figure 8 illustre un algorithme de multiplexage dans un point de couplage présent dans la matrice de commutation de la figure 4 ;
- la figure 9 illustre un algorithme de gestion du multiplexage par répartition dans le temps dans un point de couplage central présent dans la matrice de commutation de la figure 4 ; et
- 10 - la figure 10 présente un diagramme temporel des signaux véhiculés entre des FIFOs du module de commutation de la figure 2.

L'invention trouve des applications dans de nombreux domaines, et s'applique notamment dans le cadre de :

- la commutation à haut débit ;
- 15 - les applications distribuées ;
- la transmission et/ou la réception de données numériques ;
- les applications audio ;
- les réseaux d'entreprises ; et
- la transmission d'images en temps réel.

20 Un domaine d'application préférentiel de l'invention est celui des applications domestiques pour les transferts à haut débit. On décrit par la suite un exemple de système mettant en œuvre l'invention, utilisé pour interconnecter une pluralité de dispositifs audio et vidéo dans la maison.

25 Les normes IEEE 1394 et IEEE 1355 sont adaptées respectivement aux communications série et aux communications point à point.

Le système présenté dans la suite du document est un réseau fédérateur à débit commuté permettant d'atteindre des capacités de transmission élevées (typiquement de l'ordre du Gbit), et répondant au besoin important d'interfaçage des deux types de norme ci-dessus.

30 Le principe général de l'invention repose sur la mise en œuvre :

- d'un protocole d'échange ; et
- d'une matrice d'arbitrage dans un module de commutation pouvant recevoir et émettre des données de plusieurs sources, notamment via des interfaces de type IEEE1394 et/ou IEEE1355.

5 L'interface de type IEEE1355 comprend plusieurs paires de ports entrées/sorties, une paire de FIFO étant associée à chaque paire de ports.

L'interface de type IEEE1394 comprend une mémoire DPRAM permettant le stockage des données en provenance et/ou à destination d'un bus IEEE1394.

Cette DPRAM peut aussi être utilisée pour le transfert de données vers
10 et/ou à destination d'autres interfaces de communication.

Le module de commutation architecturé autour de deux bus de données (l'un en réception et l'autre en émission) de type MRT comprend notamment :

- la matrice d'arbitrage ;
- des registres trois états pilotés par des signaux générés par la matrice
15 d'arbitrage et permettant la lecture et/ou l'écriture de données sur les deux bus de données.
- des FIFOs d'entrée utilisées pour la réception des données en provenance d'une FIFO associée à un port de réception IEEE1355;
- des FIFOs de sorties utilisées pour les données à destination de la
20 DPRAM d'interfaçage avec le bus IEEE1394; et
- des FIFOs internes utilisées pour les données en provenance de la DPRAM d'interfaçage avec le bus IEEE1394.

Les FIFOs sont préférentiellement de type mémoire double ports et
peuvent ainsi être accédées simultanément en lecture et en écriture sur deux ports
25 distincts.

Les données à destination d'un port d'émission IEEE1355 sont émises directement vers la FIFO associée à ce port via le bus de transmission de données.

Selon le protocole d'échange des données, les opérations de transfert des données entre une FIFO d'entrée ou interne et une ou plusieurs FIFO de sortie
30 et/ou ports 1355, ne peuvent être effectuées que dans le cadre d'une connexion

établie au niveau des paquets. Ainsi, pour qu'une connexion soit ouverte, à la suite d'une requête en provenance d'une FIFO d'entrée ou d'une FIFO interne, il est nécessaire que chacune de la ou des FIFOs destinataires permette l'ouverture d'une connexion. De cette manière, les FIFOs destinataires ne sont jamais saturées.

La matrice d'arbitrage est constituée de lignes et de colonnes.

Chacune des colonnes est associée à un port d'entrée et est en charge de la gestion de la réception des données en provenance de ce port d'entrée.

Chacune des lignes est associée à un port de sortie et est en charge de la gestion des données destinées à ce port de sortie.

Bien entendu, les rôles des lignes et des colonnes peuvent être échangés.

La matrice est construite de telle sorte que, à un instant donné, un seul point de couplage de chaque ligne permette une connexion.

D'une manière générale, l'accès en écriture sur les bus se fait par l'intermédiaire de registres à trois états commandés par des signaux issus de la matrice d'arbitrage.

Par ailleurs, la matrice d'arbitrage génère des signaux de commande des FIFOS en écriture.

Ainsi, en résumé, la matrice d'arbitrage gère des connexions entre ports d'entrées et des ports de sortie en évitant les conflits et en gérant les aspects liés à la synchronisation. Ceci permet à la matrice d'arbitrage de générer des signaux de commande permettant l'accès (en écriture et en lecture) aux bus de transmission et de réception.

La figure 1 présente un synoptique d'un nœud de commutation 90 connecté à deux bus de communication série conformes aux normes IEEE 1394 et IEEE1355.

Le nœud représenté à la figure 1 est également connecté à un ou plusieurs autres nœuds de commutation du réseau commuté auquel il appartient.

Dans le mode de réalisation représenté sur la figure 1, on a choisi de représenter un appareil de traitement de données 92, également appelé

périphérique, qui est associé au dispositif de commutation 90.

Un nœud de commutation comprend:

- un dispositif de commutation 90 ; et
- un appareil de traitement de données 92 associé au dispositif de commutation.

5 A titre de variante, l'appareil de traitement de données peut lui-même constituer ou comprendre le dispositif de commutation 90.

Le dispositif 90 comprend :

- une unité centrale de traitement CPU 93 ;
- un moyen de stockage permanent 94 de type ROM ; et
- 10 - un moyen de stockage temporaire 95 de type RAM, associé à l'unité centrale 93, et dans lequel est chargée une architecture logicielle à l'initialisation.

Le moyen de stockage 95 est apte à stocker des paquets de données de différents types, notamment :

- des paquets asynchrones du type conforme à la norme IEEE 1394 ;
- 15 - des paquets constituant des messages en mode non connecté (asynchrones), du type conforme à la norme IEEE 1355 ;
- des paquets de contrôle du type conforme à la norme IEEE 1355 ;
- des paquets stream (isochrones) du type conforme à la norme IEEE 1355.

20 Les paquets de type conforme à la norme IEEE 1355 ont réellement une existence au niveau du composant 104 qui sera mentionné ultérieurement mais ils ne sont pas stockés sous cette forme dans le moyen de stockage RAM 95. On notera que le moyen de stockage 95 contient les informations nécessaires pour générer les paquets IEEE 1355.

25 Le cheminement de tels paquets jusqu'au moyen de stockage 95, provenant soit du bus 1394 soit du réseau commuté constitué de liens IEEE 1355 sera détaillé ultérieurement.

30 Le cheminement de tels paquets, depuis le moyen de stockage 95 jusqu'à leur transmission par le nœud de commutation, soit en direction du bus IEEE1394, soit en direction du réseau commuté constitué de liens IEEE 1355 sera détaillé ultérieurement.



Ces trois éléments 93, 94 et 95 communiquent au moyen de bus d'adresses et de données respectifs notés 96, 97 et 98, avec un bloc noté 99 et connu de l'homme du métier sous le nom de contrôleur de bus.

5 Ce bloc 99 permet notamment d'échanger des données au moyen d'un bus principal 100 avec au moins un composant d'interface de bus 101. Dans le cas où le bus 100 est un bus standard PCI (PCI signifiant en terminologie anglo-saxonne "Peripheral Component Interconnect"), le composant 101 peut être un composant dénommé AMCC 5933QC commercialisé par la société APPLIED MICRO CIRCUITS CORPORATION (Marque déposée).

10 Le bus 100 peut également connecter entre eux d'autres éléments, non représentés sur la figure 1, eux-mêmes pourvus d'une interface de bus et pouvant mettre en œuvre, par exemple, des fonctions de traitement de données.

Par exemple, dans un cas où le bus 100 est un bus standard PCI, le bloc 99 est en fait un ensemble de composants PCI tel que l'ensemble Intel 440LX AGP
15 ("Intel 440LX AGPset" dans la terminologie anglo-saxonne) commercialisé par la société INTEL (Marque déposée).

Ainsi, le bloc 99 comprend, par exemple, un composant 82443LX qui assure l'interface avec la mémoire 95 via le bus mémoire 98 et avec l'unité centrale de traitement CPU 93 via le bus local 96. Le composant 82443LX est lui-même relié à un composant 82371AB qui fournit une interface avec le bus ISA 97
20 relié à la mémoire 94. Un contrôleur d'interruption IOAPIC Intel 82093AA connecté à l'unité centrale de traitement CPU 93 gère les interruptions pouvant survenir dans le système.

Comme représenté sur la figure 1, le dispositif 90 comporte également une
25 interface de bus 102 qui peut être similaire à l'interface de bus 101 permettant ainsi à l'appareil de traitement de données ou périphérique 92 d'accéder au dispositif de communication.

Une telle interface est par exemple réalisée sous la forme d'une carte SEDNET PCI commercialisée par la société SEDERTA (Marque déposée) sous la
30 référence SD-PCI-200 et permet d'y connecter n'importe quel appareil de

traitement de données existant, conçu pour fonctionner en conformité avec la norme 1394.

Il est bien entendu possible d'utiliser un adaptateur 102 spécifique à l'appareil de traitement de données que l'on souhaite y connecter. L'adaptateur
5 102 comprend essentiellement un composant d'interface similaire au composant d'interface de bus 101.

Selon le type d'appareil de traitement de données, le bus principal 100, ainsi que les composants d'interface de bus 101 et contrôleur de bus 99 peuvent être adaptés en fonction de l'architecture du type de l'appareil. Il en est de même
10 pour l'ensemble des éléments CPU 93, RAM 95 et ROM 94.

Comme représenté sur la figure 1, le nœud selon l'invention comporte également deux moyens d'interfaçage 103 et 104.

Le moyen 103 est destiné à assurer l'interface entre le nœud 90 et le bus de communication série prévu pour fonctionner selon la norme IEEE 1394 auquel est
15 rattaché ledit nœud.

Le moyen d'interfaçage 103 est un ensemble de composants PHY/LINK 1394 qui est par exemple constitué d'un composant PHY TSB21LV03A et d'un composant LINK TSB12LV01A commercialisés par la société TEXAS INSTRUMENT (Marque déposée) et de connecteurs 1394, par exemple
20 commercialisés par la société MOLEX (Marque déposée), par exemple sous la référence 53462.

Le moyen d'interfaçage 103 comporte au moins un port externe destiné à être connecté à un appareil de traitement de données ou périphérique qui est rattaché au bus de communication série 1394.

25 Le moyen 103 comporte des moyens de comptage du nombre d'impulsions en fonction d'un signal d'horloge généré par un module de contrôle 107, qui sera défini ultérieurement. Ce signal d'horloge est synchronisé avec le "Maître de cycle" du bus de communication série avec lequel il est en relation, par l'intermédiaire des paquets de début de cycle, appelés "Cycle Start packet" en
30 terminologie anglo-saxonne. La fréquence du signal d'horloge généré par le



module de contrôle¹⁰⁷ est égale à 24,576 MHz +/- 100 ppm. Ce signal est représenté comme étant l'un des signaux notés ctrl³ sur la figure 1.

Sur chaque bus de communication série du réseau, l'un des nœuds est appelé "Maître de cycle" ("Cycle Master" en terminologie anglo-saxonne) et le nœud "Maître de cycle" du bus "racine" est appelé "Maître de cycle du réseau" ("Net Cycle Master" en terminologie anglo-saxonne).

En outre, tous les nœuds "Maîtres de cycle" du réseau présentent une caractéristique qui leur est propre, puisqu'elle dépend de la fréquence de leur horloge interne, à partir de laquelle est définie la durée d'une "période de référence" ou "cycle".

La durée du cycle notée T est égale à un nombre entier n_{int} d'impulsions d'horloge, commun ou non à tous les bus, et qui est multiplié par l'inverse de la fréquence de l'horloge interne propre au nœud "Maître de cycle".

La durée du cycle T est ainsi, par exemple, égale à 125 microsecondes.

Lorsque deux bus de communication série sont reliés par un pont, le "Maître de cycle" de l'un des bus doit synchroniser ses cycles par rapport aux cycles générés par le "Maître de cycle" du bus adjacent.

D'une manière générale, les réseaux de communication formés de bus de communication série permettent la transmission de paquets synchronisée à partir des cycles de bus considérés. Les bus sont par exemple utilisés pour transmettre des paquets de données en temps réel du type audio/vidéo.

Les moyens de comptage comme ceux du moyen d'interfaçage 103 cité plus haut se présentent par exemple sous la forme d'un registre.

Le moyen d'interfaçage 104 mentionné ci-dessus est un composant d'interface IEEE 1355 qui comporte trois ports. Il comprend notamment un composant C113 commercialisé par la société 4LINKS ainsi que trois composants d'interface LUC1141MK commercialisés par la société LUCENT (Marque déposée), eux-mêmes reliés à des connecteurs IEEE 1355, par exemple commercialisés par la société HARTING (Marque déposée). Le composant C113 est lui-même réalisé sur la base d'un composant programmable de type FPGA

("Field Programmable Gate Array" en terminologie anglo-saxonne) Spartan XCS30XL, commercialisé par la société XILINX (Marque déposée).

Les initiales FPGA correspondent approximativement en français à "Matrice de Portes Programmables".

5 Les trois ports externes du moyen d'interfaçage 104 sont destinés à être connectés à des ports de même type sur un autre nœud de commutation du réseau commuté, permettant ainsi au dispositif 90 de communiquer avec un autre nœud de ce réseau.

10 Le dispositif 90 comporte également un moyen de contrôle de flux de données 105 qui permet le transfert des données entre les différents composants d'interface 101, 103 et 104. Ce moyen 105 est réalisé en logique programmable, exécuté par un composant de type FPGA, par exemple de référence VIRTEX, commercialisé par la société XILINX.

15 Ce moyen 105 met en œuvre notamment une unité de mémorisation à double port 106 qui permet de stocker des données à destination de, ou provenant du réseau commuté 1355.

L'unité de mémorisation à double port possède une capacité de stockage inférieure à 2 Mbits et est, par exemple, réalisée sous la forme d'une mémoire de type DPRAM à accès 32 bits.

20 Les initiales DPRAM signifient en terminologie anglo-saxonne "Dual Port Random Access Memory" ce qui peut être approximativement traduit en langue française par "Mémoire volatile à double port".

25 L'unité de mémorisation 106 comporte une pluralité de zones mémoires qui sont gérées comme des mémoires individuelles de type FIFO, initiales des termes anglais "First-in First-out" signifiant en français "Premier entré Premier sorti".

Une telle zone mémoire correspond à une mémoire dans laquelle les données sont lues dans l'ordre dans lequel elles ont été préalablement écrites.

30 Ces zones mémoires comportent chacune un pointeur de lecture et un pointeur d'écriture associés.



Chaque zone mémoire étant gérée comme une mémoire de type FIFO, son remplissage et son vidage peuvent s'effectuer en même temps, et de manière indépendante. Ceci permet de désynchroniser les opérations de lecture et d'écriture des données, effectuées par une unité de commutation 108 qui sera
5 définie ultérieurement, des opérations de lecture et d'écriture des données, effectuées par le module de contrôle 107.

En effet, le taux d'occupation de la zone mémoire considérée est géré de manière circulaire et l'on sait à tout moment si les données contenues dans une zone mémoire ont été lues ou non. Lorsque ces données ont été lues, il est alors
10 possible de venir écrire de nouvelles données à la place de celles-ci.

L'unité de mémorisation à double port constitue en quelque sorte une file d'attente pour les paquets, et la fonction de stockage est réalisée de manière indépendante, selon le port par lequel les paquets parviennent à l'unité de mémorisation.

15 D'une manière générale, toutes les données isochrones ou asynchrones provenant du réseau commuté sont stockées dans l'unité de mémorisation 106.

Ce stockage est temporaire pour les paquets de données asynchrones (paquets constituant un message transmis en mode non connecté) et pour les paquets de contrôle, qui sont amenés à être transférés ensuite dans le moyen de
20 stockage RAM 95 pour un stockage d'une durée plus importante.

En revanche, les paquets de données isochrones (paquets de type "stream", c'est-à-dire transmis en mode connecté), sont stockés uniquement dans cette unité de mémorisation 106 avant leur transmission sur le bus de communication auquel est raccordé le nœud de commutation 90 ou sur le réseau commuté.

25 Ceci s'explique par le fait que ce type de données doit être transféré aussi rapidement que possible du réseau commuté vers le bus et donc doit être stocké dans un moyen de stockage facilement et rapidement accessible.

De même, les paquets de données isochrones, issus du bus de communication auquel est raccordé le nœud de commutation 90, et qui sont
30 destinés au réseau commuté, sont stockés uniquement dans l'unité de

mémorisation 106, et non dans le moyen de stockage 95, pour les mêmes raisons que celles invoquées précédemment.

Ainsi que représenté sur la figure 1, le moyen de contrôle de flux de données 105 comporte plusieurs autres éléments dont un module de contrôle 107 (déjà mentionné plus haut), qui assure une fonction de contrôle de l'unité de
5 mémorisation 106, une unité de commutation 108 (déjà mentionnée plus haut) en communication avec le moyen d'interfaçage 104, avec l'unité de mémorisation 106 et avec le module de contrôle 107, ainsi qu'une unité d'ordonnancement des paquets de données 109, qui est relation avec le module de contrôle 107.

10 On notera également que le module de contrôle 107 communique avec les moyens d'interfaçage 103 et 104 ainsi qu'avec le composant d'interface de bus noté 101.

Le module de contrôle 107 a pour fonction de multiplexer les accès en lecture ou en écriture à des registres d'autres modules à partir du bus principal
15 noté 100.

Le module 107 possède également la maîtrise du composant d'interface de bus 101 pour les opérations de lecture et d'écriture sur le bus principal 100, incluant notamment le transfert en "mode rafale" (connu en terminologie anglo-saxonne sous le terme de "burst mode").

20 Le module de contrôle 107 est également chargé du déclenchement des interruptions sur le bus principal 100, en fonction d'événements de communication particuliers.

Ce module échange des données avec le composant 101, sur un bus additionnel 110 (connu en terminologie anglo-saxonne sous le terme de "add-on
25 bus"), suivant les signaux de contrôle notés *ctrl1*.

Comme annoncé ci-dessus, le module 107 est chargé du contrôle de l'unité de mémorisation 106, en ce qui concerne les opérations de lecture et d'écriture en mode FIFO, dans le cas particulier où le composant d'interface de bus 101 est un AMCC, par l'intermédiaire d'un bus de données 111 et de signaux de contrôle
30 *ctrl2*.



Le moyen d'interfaçage 103 contient des mémoires de type FIFO, qui sont utilisées lors du transfert de paquets de données de type conforme à la norme IEEE 1394. Il comprend deux mémoires FIFO de transmission dites ATF ("Asynchronous Transfer FIFO" en terminologie anglo-saxonne) et ITF 5 ("Isochronous Transfer FIFO" en terminologie anglosaxonne) et une mémoire FIFO de réception dite GRF ("General Receive FIFO" en terminologie anglo-saxonne). Ces mémoires FIFO sont plus largement décrites dans la documentation associée au composant LINK TSB12LV01A.

Le module de contrôle 107 et le moyen d'interfaçage 103 gèrent le transfert 10 de données sur un bus 112 suivant des signaux de contrôle *ctrl3*.

Par ailleurs, le module de contrôle 107 contrôle l'unité de commutation 108, au moyen de signaux de contrôle *ctrl 4*, afin de transférer des données de l'unité de commutation vers l'unité de mémorisation 106 par l'intermédiaire d'un bus de données 113, et inversement.

15 L'unité de commutation 108 est connectée au moyen d'interface 104 par l'intermédiaire d'un bus de données 114 et de signaux de contrôle *ctrl5*.

L'unité d'ordonnancement des paquets de données 109, notée également SAR (connue en terminologie anglo-saxonne sous le terme de "Segmentation And Reassembling"), informe le module de contrôle 107 du ou des prochains paquets 20 de données à transmettre, par l'intermédiaire de signaux de contrôle *ctrl6*.

En outre, l'unité d'ordonnancement 109 vérifie la réception des paquets de données, et gère l'allocation et la libération de zones mémoires (connues en terminologie anglo-saxonne sous le terme de "buffers") de l'unité de mémorisation 106.

25 Les signaux de contrôle *ctrl7* échangés entre le moyen d'interfaçage 104 et le module de contrôle 107 comprennent notamment les signaux d'horloges régénérés à partir de la réception des paquets 1355 sur chacun des trois ports du moyen d'interfaçage 104.

Un quartz à 49,152Mhz (non représenté) est connecté à la fois au moyen 30 104 pour l'émission des paquets 1355 et au module de contrôle 107 qui génère un

signal d'horloge à 24,576 MHz +/- 100 ppm, d'une part, pour l'unité d'ordonnancement des paquets de données 109, afin de cadencer l'émission des paquets 1355 et, d'autre part pour le moyen d'interfaçage 103, afin de cadencer l'émission des paquets 1394.

5 La figure 2 présente un synoptique détaillé du moyen de contrôle de flux de données 105, de l'interface 104 illustrés en regard de la figure 1, ainsi que leurs interconnexions.

Quelques détails sur l'interface série 104 permettent de comprendre comment des paquets sont multiplexés sur le bus de données 114.

10 L'interface série 104 comprend notamment :

- un moyen de contrôle de FIFO 120 décrit précédemment en regard de la figure 1 ;
- trois FIFOs d'entrée 121 (InPort0_FIFO), 122 (InPort1_FIFO) et 123 (InPort2_FIFO);
- 15 - un multiplexeur 127 ; et
- des FIFO de sorties 124 (OutPort0_FIFO), 125 (OutPort1_FIFO) et 126 (OutPort2_FIFO).

Les FIFO 121, 122 et 123 sont utilisées pour la mémorisation de paquets entrants, alors que les FIFO 124 à 126 sont utilisées pour la mémorisation des paquets sortants.

Une paire de FIFO est reliée à chaque port de l'interface 104 IEEE 1355.

Ainsi, par exemple :

- les FIFO 121 et 124 sont utilisées pour le port 0 ;
- les FIFO 122 et 125 pour le port 1 ; et
- 25 - les FIFO 123 et 126 pour le port 2.

Le bus de données 114, illustré en regard de la figure 1 se décompose en deux bus sur la figure 2 :

- un bus 44 $RxDI[17:0]$ de liaison en réception des données destinées à l'unité de commutation 108 et transmises par l'interface 104 ; et
- 30 - un bus 41 $TxDO[17:0]$ de liaison en émission des données à partir de l'unité



de commutation 108, vers l'interface 104.

En variante, les bus 44 et 41 forment un seul bus.

Après leur écriture dans l'une des FIFO d'entrée 121, 122 et 123, les paquets entrants sont multiplexés à travers le multiplexeur 127 avant d'être renvoyés vers l'unité de commutation 108 à travers le bus 44.

Les données sortantes de l'unité de commutation 108 sont émises à travers le bus de données 41 pour être stockées dans l'une des FIFO de sortie 124, 125 ou 126.

On rappelle que le moyen de contrôle de flux de données 105 comprend notamment :

- le module de contrôle 107 ;
- l'unité de commutation 108 ; et
- l'unité de mémorisation à double port ou DPRAM 106.

Le signal de contrôle *Ctrl5*, illustré en regard de la figure 1 se décompose en trois signaux sur la figure 2 :

- un signal de contrôle 51 ;
- un signal de contrôle 43 *TxFIFOfull[0 :2]* ; et
- un signal de contrôle 42 *WriteTXFIFO[0 :2]*.

Les signaux de contrôle 51, *TxFIFOfull[0 :2]* 42 et *WriteTXFIFO[0 :2]* 43 permettent au moyen de contrôle 120 de la FIFO de gérer le transfert des paquets de données entre l'unité de commutation 108 et l'interface 104 de liaison série, à travers les bus de données 44 et 41.

Le traitement des signaux de contrôle dans l'unité de commutation 108 sera décrit ultérieurement en regard de la figure 3.

Le signal *Ctrl4*, illustré en regard de la figure 1, entre l'unité de commutation 108 et le module de contrôle 107, se décompose en au moins quatre signaux sur la figure 2 :

- deux signaux de contrôle 53 et 54 ;
- un signal de contrôle *IntRxTxEOP[1 :0]* 45a ;
- un signal de données *IntRx_TX[31 :0]* 45b.

Le signal 45b est un bus de données reliant les éléments 106, 107 et 108.
Le module de contrôle lit les en-têtes paquets à partir de ce bus 45b.

Les signaux de contrôle 53, 54 et 45a permettent la gestion de transfert de données *IntRx_TX[31:0]* 45b entre la mémoire double port 106 et l'unité de
5 commutation 108.

Dans certains cas, notamment lors du traitement d'en-tête de paquet, les données du paquet sont envoyées au module de contrôle 107 plutôt qu'à l'unité de mémorisation 106 pour une analyse ultérieure.

Le bus de données 45b est représenté sur la figure 1 par le bus de données
10 113.

La figure 3 représente un schéma électronique du module de commutation 108 de la figure 2.

Le module de commutation 108 comprend notamment :

- un moyen 60 d'arbitrage ;
- 15 - un moyen 20 de génération de cycle ;
- des moyens chargés de la réception et de l'émission des données en provenance ou à destination de l'interface 104 (soient les données en provenance ou à destination d'un réseau commuté IEEE1355) ; et
- des moyens chargés de la réception des données en provenance de la mémoire
20 DPRAM 106 (soient les données en provenance d'un bus IEEE1394) ; et
- des moyens chargés de l'émission des données vers la mémoire DPRAM 106 (soient les données à destination d'un bus IEEE1394) ;

Les moyens chargés de la réception et de l'émission des données en provenance de ou vers l'interface 104 comprennent notamment :

- 25 - un moyen 18 de modification d'en-tête ;
- un moyen 19 d'analyse d'en-tête en réception ;
- deux mémoires volatiles 15 et 16 de réception agencées en FIFO ; et
- un moyen 17 de contrôle des FIFOs de réception 15 et 16.

Les moyens chargés de la réception des données en provenance de la
30 mémoire DPRAM 106 comprennent notamment :



- quatre FIFOs d'entrée 9, 10, 11 et 12 ;
- un moyen 130 de contrôle des FIFOs d'entrée ;
- un multiplexeur 50 ; et
- un moyen 13 d'analyse d'en-tête de transmission.

5 Les moyens chargés de l'émission des données vers la mémoire DPRAM 106 comprennent notamment :

- trois mémoires d'assemblage 22, 23 ou 24 ;
- trois FIFOs de sortie 25, 26 et 27 ;
- un moyen de contrôle 84 des FIFOs de sortie ; et

10 - un multiplexeur 28.

Ces éléments faisant partie du module de commutation 108 sont reliés par des éléments de liaisons décrits ci-après. On note néanmoins que la plupart des échanges se font autour du moyen 60 d'arbitrage avec utilisation de deux bus de données principaux, *TxBus* 70 et *RxBus* 80 dont les accès sont gérés par le moyen d'arbitrage 60.

L'utilisation des deux bus internes, les bus de transmission *TxBus* 70 et de réception *RxBus* 80, qui permettent de traiter les transferts de données entre les ports va maintenant être détaillée.

20 Les paquets sortants vers le bus 41 sont transmis à travers le bus de transmission *TxBus* 70.

Les paquets sortants vers le bus 45 sont transmis à travers le bus interne de réception *RxBus* 80.

Le moyen 60 d'arbitrage est en charge de la gestion des accès en écriture sur le bus *TxBus* 70 contrôlant l'ouverture :

- 25
- de registres à trois états 1, 2, 3 et 4 grâce à des signaux *OpenTxTS[0:3]_onRx* 40 pour les données issues des FIFO d'entrée 9, 10, 11 et 12 ; et
 - de registres à trois états 14 grâce à des signaux de contrôle *OpenRxTS_onTx* 33 pour les données issues de la mémoire 16.

30 Les opérations de lecture du bus *TxBus* 70 vers un port de sortie de l'interface 104 sont gérées par le moyen d'arbitrage 60 en fonction du signal de

contrôle vers les FIFO associées à chaque port *TxFIFOfull[0:2]* 43 et *WriteTxFIFO[0:2]* 42.

Le moyen d'arbitrage 60 gère aussi les accès en écriture vers le bus *RxBus* 80 en contrôlant l'ouverture :

- 5 - de registres à trois états 5, 6, 7 et 8 à travers des signaux *OpenTxTs[0:3]_onRx* 39 pour les données issues des FIFO d'entrée 9, 10, 11 et 12 ; et
- d'un registre à trois états 21 à travers des signaux de contrôle *OpenRxTs_onRx* 34 pour les données issues de la mémoire 15.

10 Le moyen d'arbitrage 60 effectue aussi les opérations de lecture de données sur le bus *RxBus* 80 pour une écriture vers l'une des mémoires d'assemblage 22, 23 ou 24 (en anglais "assembly buffers"). L'"assembly buffer" sert de tampon pour la conversion des données entre le bus 80, sur lequel les données sont exprimées sur 18 bits (16 bits de données et 2 bits de contrôle), et les

15 O_FIFOs 25, 26 et 27, dans lesquelles les données sont exprimées sur 34 bits (32 bits de données et 2 bits de contrôle).

Ces tailles de bus sont liées à des contraintes physiques. Le module 84 transmet au moyen d'arbitrage 60 un signal de contrôle *Rx FIFOfull[0:2]* 83, indiquant le degré de remplissage des FIFO 25, 26 et 27. Le moyen d'arbitrage 60

20 génère un signal de contrôle *WriteO_FIFO[2:0]* 35 à destination d'un module 84 de contrôle des FIFO de sortie, en fonction du degré de remplissage des FIFOS 25, 26 et 27.

Le moyen d'arbitrage 60 pour le multiplexage des données sur les bus 70 de transmission *TxBus* et 80 de réception *RxBus* utilise des signaux 46 de cycle

25 *EN_cycle[0:2]*. C'est le moyen 20 de génération de cycle, par exemple un compteur de Johnson, qui génère ces signaux 46.

Le traitement des données entrantes va maintenant être détaillé en fonction de l'origine de ces données :

- le bus 44 ;
- 30 - le bus 45.



Les données entrantes en provenance du bus 44 *RxDI[17:0]* ou du bus 45 seront routées vers le commutateur pour aller :

- vers le bus 41 *TxD0[17:0]* soit en direction de l'interface 104 ; ou
- vers le bus 45 soit en direction de la DPRAM 106.

5 Le traitement des données entrantes en provenance du bus 44 va maintenant être exposé.

Les échanges de données sur le bus 44 sont gérés à travers le moyen 18 de modification d'en-tête et le moyen de contrôle de FIFO 120 de l'interface série, grâce aux signaux de contrôle 51. Les signaux de contrôle 51 comprennent un
10 signal d'écriture vers le moyen 18 de modification d'en-tête et des signaux de contrôle de flux du module 18, correspondant chacun à des ports d'entrées du moyen d'interfaçage 104.

Les morceaux de paquets entrant en provenance du bus 44 parviennent d'abord à un moyen de modification d'en-tête 18 pour un traitement relatif aux
15 en-têtes de paquet.

Les morceaux de paquets sont ensuite stockés :

- dans la mémoire 15 lorsqu'ils sont adressés au bus de réception 80 *RxBus* ; et
- dans la mémoire 16 lorsqu'ils sont adressés au bus de transmission
20 70 *TxBus*.

Lorsqu'ils sont adressés vers les deux bus, les morceaux de paquets sont adressés aux deux bus simultanément pour être stockés dans deux moyens de stockage simultanément.

De même, le moyen 17 de contrôle de FIFO de réception contrôle les
25 opérations de stockage dans les deux mémoires 15 et 16.

Les mémoires 15 et 16 sont des mémoires à double port, découpées en trois zones indépendantes, chacune étant gérée comme une FIFO.

Les signaux synchrones sont gérés suivant un certain rythme correspondant à une succession de phases constituées de quatre cycles d'horloge.

Dans une phase à quatre cycles, au cours de chacun des trois premiers cycles d'horloge de la phase, on effectue à la fois une opération d'écriture et une opération en lecture vers deux adresses de chacune des RAM (mémoires volatiles) 15 ou 16. Ces adresses pointent respectivement :

- 5 - vers le dernier élément (queue) de l'une des FIFO dans chacune des RAM 15 et 16 ; et
- vers le premier élément (head) de l'une des FIFO dans chacune des RAM 15 et 16.

Aucune opération de lecture ou d'écriture vers les RAM 15 et 16 n'est effectuée au cours du quatrième cycle d'horloge dans une phase à quatre cycles.

Chaque opération de lecture effective de FIFO (définies lorsque le signal 32 de sortie *Rx[i]_read* est activé durant un cycle et est suivi par un signal d'entrée *AllData used[i]* activé durant un cycle, *i* correspondant à un numéro de la FIFO d'entrée) modifie l'adresse pointant sur l'en-tête de FIFO.

15 Des opérations effectives d'écriture de FIFO (basées sur le même mécanisme que les opérations de lecture, utilisant ici le signal 52 illustré en regard de la figure 3) modifient l'adresse pointant sur la queue de la FIFO.

Ainsi, durant trois cycles, chaque partie de FIFO correspondant à chacun des ports d'entrée est traitée : les opérations d'écriture de données par les moyens 20 18 de modification d'en-tête sont séquentiellement adressées vers les parties correspondantes 15 et 16, chacun des morceaux de paquets entrant de chacun des ports d'entrée 121, 122 et 123 de l'interface 104.

D'une manière similaire, les données lues à partir des mémoires 15 et 16 sont traitées respectivement par les bus *RxBus* 80 et *TxBus* 70.

25 Le moyen 17 de contrôle de FIFO de réception est en charge de la lecture et de l'écriture effective des FIFO contenues dans les mémoires 15 et 16.

Le signal 52 de contrôle informe le moyen de modification d'en-tête 18 qu'une FIFO est pleine dans l'une des mémoires 15 ou 16 et qu'il y a aussi une requête d'écriture pour une FIFO dans l'une des mémoires 15 ou 16.

30 Le moyen de contrôle de FIFO de réception informe le module 60

d'arbitrage qu'une nouvelle donnée a été lue à partir de l'une des mémoires 15 ou 16 grâce au signal *Rx[0:2]read* 32.

Les mêmes données sont lues à chaque phase de quatre cycles lorsqu'elles ne sont pas lues de manière effective en les plaçant simplement, soit sur le bus *RxBus*, soit sur le bus *TxBus*, soit sur les deux, en fonction du routage du paquet.

Ainsi, les mêmes données sont présentées tous les quatre cycles, jusqu'à ce qu'elles aient été réellement lues par la ou les destinations.

Le moyen 60 d'arbitrage gère l'ouverture du registre à trois états 14 grâce aux signaux de contrôle *OpenRxTs_onTx* 33, pour placer les données en lecture de la mémoire 16 sur le bus de transmission *TxBus* 70.

Le moyen 60 gère également l'ouverture du registre à trois états 21, grâce aux signaux de contrôle *OpenRxTs_onRx* 34 pour mettre les données lues de la mémoire 15 vers le bus de réception *RxBus* 80.

Le moyen d'arbitrage 60 gère aussi le contrôle des signaux *allRxData_used[0:2]* pour indiquer au moyen 17 de contrôle des FIFO de réception que la prochaine donnée sera lue par une partie correspondante dans l'une des mémoires 15 ou 16.

Le moyen 19 d'analyse d'en-tête de paquets reçus commande la forme du signal 47 vers le moyen 60 d'arbitrage.

Lorsqu'un nouveau paquet en provenance de l'un des ports d'entrée 121, 122 ou 123 est traité dans le moyen 18 de modification d'en-tête, le signal 47 *Rx[0:2]Want[0:3]* permet la requête d'une connexion :

- vers l'un des ports de sortie de l'interface 104 à travers le bus multiplexé 41 ;
- ou
- vers l'une des FIFO internes 25, 26 ou 27 à travers le bus multiplexé 45.

Lorsque la fin du paquet est traitée dans le moyen 18 de modification d'en-tête, le signal 47 *RxEOP[0:2]* informe le moyen d'arbitrage 60 de la fin d'une connexion.

On note que le moyen 18 de modification d'en-tête émet un signal représentatif d'une information de routage à destination du moyen 19 d'analyse d'en-tête en réception.

En ce qui concerne la connexion et la déconnexion vers l'interface série IEEE 1355, le moyen 60 d'arbitrage gère les signaux *Rx[0:2]_connected* 81 émis vers le moyen 17 de contrôle de FIFO de réception pour indiquer le statut de la connexion associée à chaque port.

De même, le moyen 60 d'arbitrage gère les signaux *Tx[0:2]_connected* 82 vers le moyen 130 de contrôle des FIFO d'entrée 9, 10, 11 et 12 pour les données en provenance de la DPRAM 106.

Les données échangées sur le bus 45 sont gérées :

- par le moyen 84 de contrôle des FIFO de sortie *O_FIFO* et le module de contrôle 107 par l'intermédiaire du signal 53 ; et
- par le moyen 13 de contrôle des FIFO d'entrée *I_FIFO* et le module de contrôle 107 par l'intermédiaire du signal 54.

Le multiplexeur 28 autorise la sélection d'une FIFO de sortie *O_FIFO* parmi les trois FIFO 25, 26 ou 27 pour des opérations de lecture de FIFO de données destinées à la mémoire de stockage double port 106. Cette opération de sélection est gérée par le moyen 84 de contrôle de FIFO de sortie *O_FIFO* grâce à un signal 85.

Par ailleurs, les moyens 130 de contrôle de FIFO d'entrées *I_FIFO* contrôlent des opérations d'écriture de la mémoire de stockage 106 vers l'une des quatre FIFO *I_FIFO* d'entrée 9, 10, 11 ou 12 grâce aux signaux 90.

C'est uniquement le cas pour le bus 42 où les signaux de contrôle 43 et 42 attachés au moyen de mémorisation des ports de sorties sont directement connectés vers les moyens 60 d'arbitrage.

En d'autres termes, le bus 41 est le seul bus dont les signaux de contrôle (42 et 43) sont directement reliés à l'arbitre 60.

Le traitement des données entrantes en provenance du bus 45 va maintenant être détaillé.

Les morceaux de paquets provenant du bus 45 sont démultiplexés vers chaque FIFO d'entrée 9, 10, 11 et 12. Ainsi, quatre paquets (un par FIFO d'entrée) peuvent être traités simultanément.

Chaque FIFO d'entrée est connectée à deux bus internes 70 et 80 à travers
5 les registres à trois états 1 à 8.

Le moyen 60 d'arbitrage est en charge de gérer :

- les accès en écriture sur le bus *TxBus* 70 contrôlant alternativement l'ouverture des registres trois états, 1 à 4 grâce aux signaux *OpenTxTs[0:3]_onTx* 40, chacun des registres 1 à 4 étant affecté à une FIFO
10 d'entrée 9 à 12 ; et
- les accès en écriture sur le bus *RxBus* 80 contrôlant alternativement l'ouverture des registres trois états 5 à 8 grâce aux signaux *OpenTxTs[0:3]_onRx* 39, chacun des registres 5 à 8 étant affecté à une FIFO d'entrée 9 à 12 .

15 Les opérations d'écriture sur le bus *RxBus* 70 sont indépendantes des opérations d'écriture sur le bus *TxBus* 80.

Le moyen d'arbitrage 60 génère aussi des signaux de contrôle *allTxData used[0:3]* 37, qui autorisent la lecture sur l'une des FIFO d'entrées 9 à 12, grâce aux moyens de contrôle des FIFO d'entrée 130.

20 Le moyen 130 de contrôle des FIFO d'entrée informe le moyen 60 d'arbitrage de la réussite d'une opération de lecture, effectuée par l'une des FIFO d'entrée, grâce au signal de contrôle 38 *Tx[0:3]_Read*.

Les données d'en-tête de paquet sont multiplexées par un multiplexeur 50 de l'une des FIFO de sorties 9 à 12 pour être émises vers le moyen 13 d'analyse
25 d'en-tête.

Le moyen 13 d'analyse d'en-tête de paquet transmis analyse les données issues des FIFO d'entrée 9, 10, 11 et 12 pour générer le signal 36 *Tx[0:3]Want[0:3]* à destination du moyen d'arbitrage 60.

Lorsqu'un nouveau paquet est traité dans l'une des FIFO d'entrée 9 à 12,
30 le signal 36 *Tx[0:3]Want[0:3]* permet la requête d'une connexion :

- vers l'un des ports de sortie du moyen d'interfaçage 104 par un bus 41 multiplexé ; ou
- vers l'une des FIFO de sortie 25, 26 ou 27 par un bus 45 multiplexé.

5 Lorsque la fin du paquet est traitée dans l'une des FIFO d'entrée 9 à 12, le signal 36 *TxEOP[0:3]* permet d'informer le moyen d'arbitrage 60 d'une fin de connexion.

La figure 4 illustre une implémentation détaillée du moyen 60 d'arbitrage. Le moyen d'arbitrage est en charge de la génération du multiplexage par répartition dans le temps (MRT) en combinaison avec la gestion des connexions.

10 Ici, une connexion se réfère à un intervalle alloué à un paquet de routage d'un port entrant vers un port sortant.

Un cycle de connexion est associé à chaque port sortant. L'établissement d'une connexion permet d'associer un port entrant à au moins un cycle de connexion et de valider le mécanisme d'écriture des unités de données de

15 protocole pour chaque donnée élémentaire vers le port sortant associé au(x) cycle(s) de connexion considéré(s).

L'élément clé du moyen 60 d'arbitrage est un point de couplage élémentaire.

Dans le mode préféré de réalisation, le moyen 60 d'arbitrage comprend

20 notamment :

- 30 points de couplage élémentaires tels qu'illustrés en regard de la figure 4 et référencés 150 à 179 ; et
- 7 moyens 180 à 186 de traitement central de points de couplage effectuant des opérations de type « OU » sur des signaux élémentaires ;
- 25 et
- un moyen de traitement 187 effectuant des opérations de type « OU » sur des signaux élémentaires.

Les 30 points de couplage élémentaires sont organisés en lignes et colonnes. Les colonnes traitent des signaux attachés aux ports entrants alors que

30 les lignes traitent des signaux attachés aux ports sortants.

Une colonne est allouée à chaque FIFO d'entrée ou d'une manière équivalente à chaque paquet entrant :

- Ainsi, les première, seconde et troisième colonnes correspondent aux trois FIFO réalisées à partir de mémoires double port 15 et 16 telles qu'illustrées en regard de la figure 3), respectivement attachées aux FIFO InPort0_FIFO 121, InPort1_FIFO 122 et InPort2_FIFO 123 de l'interface série 104 .
- De même, les 4^{ème}, 5^{ème}, 7^{ème} et 8^{ème} colonnes sont respectivement attachées aux FIFO d'entrées I_FIFO_0 9, I_FIFO_1 10, I_FIFO_2 11 et I_FIFO_3 12.

Une ligne est allouée à chaque FIFO de sortie ou d'une manière équivalente attachée à chaque paquet sortant :

- Ainsi, les première, seconde, et troisième lignes correspondent aux trois FIFO de sortie OutPort0 FIFO 124, OutPort1 FIFO 125 et OutPort2 FIFO 126 avec l'interface série et attachés aux intervalles de temps du bus *TxBus* 70.
- De même, les 4^{ème}, 5^{ème}, 6^{ème} lignes correspondent aux FIFO internes de sortie du commutateur, respectivement les FIFO O_FIFO_0 25, O_FIFO_1 26 et O_FIFO_2 27 et attachés aux intervalles de temps du bus *RxBus* 80.

Les intervalles disponibles pour la connexion sont relatifs à des lignes (donc à des ports de sortie), ce qui signifie qu'il est possible de traiter six connexions simultanément. Tant qu'un point de couplage a été sélectionné avec une ligne, l'intervalle de temps correspondant ou la FIFO de sortie correspondante sont alloués à un paquet entrant. Ainsi, aucun autre paquet ne peut être transmis au cours de cet intervalle de temps alors que la connexion existante reste active, et ce jusque la fin du transfert du paquet courant.

Pour obtenir un tel comportement, les points de couplage élémentaires sur une ligne sont chaînés en cascade afin de permettre une seule connexion à la fois. Par exemple, les points de couplage élémentaires 150 à 154 sont chaînés en

cascade pour permettre le transfert d'un seul paquet entrant au cours de l'intervalle 0 du bus *TxBus* 70. De la même manière, les points de couplage élémentaires 165 à 169 sont chaînés en cascade pour permettre le transfert d'un seul paquet entrant au cours de l'intervalle 0 du bus *RxBus* 80. Ces chaînages en cascade sont détaillés en regard de la figure 5.

Selon le mode préféré de réalisation, un paquet entrant peut être transféré simultanément sur quatre intervalles lorsque la matrice d'arbitrage comprend quatre points de couplage élémentaire par colonne distincte de la dernière colonne.

La dernière colonne permet le transfert des paquets vers tous les intervalles simultanément et la matrice d'arbitrage comprend six points de couplage élémentaires associés à cette dernière colonne. Cette dernière colonne correspond à la mise en œuvre particulière de la FIFO de diffusion I_FIFO_3 12.

Selon un mode particulier de réalisation du commutateur, une matrice d'arbitrage simplifiée permet la suppression des couples élémentaires de couplage et par exemple lorsque le bus 45 est attaché à un point interne, les points de couplage 166 à 169, 171 à 174 et 176 à 179 ne sont plus utiles, car ils ne seraient utilisés que pour du trafic généré par le point interne pour lui-même.

Selon le mode préféré de réalisation, pour chaque signalisation de colonne, les signaux d'entrée 471a à 471c et 47a à 47c (*Rx[0:2]EOP* et *Rx[0:2]Want[0:3]*) et 361a à 361d et 36a à 36d (*Tx[0:2]EOP* et *Tx[0:2]Want[0:3]*) informent chaque point de couplage élémentaire de la même colonne, respectivement de la fin du transfert d'un paquet (libération de la connexion) ou d'un nouveau paquet à transférer (demande de connexion).

On note que les signaux d'entrée 361a à 361d et 36a à 36d (tels qu'illustrés en regard de la figure 3 sous la référence 36) sont générés par le moyen 13 d'analyse d'en-tête de transmission, sur réception d'un marqueur de fin de paquet (EOP).

Les signaux d'entrée 32a à 32c (*Rx[0:2]_read*) et 38a à 38d (*Tx[0:2]_read*) informent chaque point de couplage élémentaire de la même



colonne, de la disponibilité d'une nouvelle donnée élémentaire à traiter pendant le cycle de connexion préalablement déterminé au cours de l'établissement de cette connexion.

Les signaux 471a à 471c et 47a à 47c sont générés par le moyen 19
5 d'analyse d'en-tête de réception, tel qu'illustré en regard de la figure 3 sous la référence 47, sur réception d'un marqueur de fin de paquet (EOP).

Les signaux 32a à 32c et 38a à 38d sont générés respectivement par les
moyens 17 de contrôle de FIFO de réception et moyens 130 de contrôle de FIFO
d'entrée tels qu'illustrés en regard de la figure 3 sous les références respectives 32
10 et 38.

Chaque colonne est associée à l'un des points de couplage central 180 à 186. Ainsi, par exemple, la colonne comprenant les points de couplage élémentaires 150, 155, 160 et 165 est associée au point de couplage central 180.

Chacun des moyens 180 à 186 de traitement effectue une opération de type
15 « OU » sur les signaux de contrôle générés pour chaque intervalle correspondant à la colonne qui lui est associée et pour chaque bus (respectivement sur les bus *TxBus* 70 et *RxBus* 80), correspondant à chaque cycle de connexion.

Un signal de sortie, 203 à 209, issus de chaque point de couplage central respectivement 180 à 186 est connecté à chaque point de couplage de la colonne
20 correspondante au point de couplage considéré et permet de valider l'écriture effective de l'unité de protocole vers le port sortant associé au cycle de connexion considéré.

Les signaux de sorties 203 à 209 représentent respectivement les signaux :

- *OpenRxTs0_onTx[0:2]* et *Rx0_EOP_sent* (203) ;
- 25 - *OpenRxTs1_onTx[0:2]* et *Rx1_EOP_sent* (204) ;
- *OpenRxTs2_onTx[0:2]* et *Rx2_EOP_sent* (205) ;
- *OpenTxTs0_onTx[0:2]* et *Tx0_EOP_sent* (206) ;
- *OpenTxTs1_onTx[0:2]* et *Tx1_EOP_sent* (207) ;
- *OpenTxTs2_onTx[0:2]* et *Tx2_EOP_sent* (208) ; et
- 30 - *OpenTxTs3_onTx[0:2]* et *Tx3_EOP_sent* (209).

Le moyen 187 effectue une opération de type « OU » (représenté par le signe \oplus) sur les signaux de sortie 216 à 218 (en respectant l'implémentation spécifique des FIFO présentes dans les mémoires double port 15 et 16) des points 180 à 182 de couplage central qui correspondent aux trois premières colonnes de la matrice d'arbitrage.

Ainsi, par exemple, une donnée élémentaire présente dans l'une des FIFOs de la mémoire RAM 15 est présentée sur le bus *RxBus* 80 lorsque l'un des signaux *OpenRxTs0_onRx[0:2]* active l'ouverture du registre 21 à trois états associé à la FIFO de la mémoire 15. Comme l'ensemble des trois FIFO de la mémoire 15 associées aux FIFO InPort0_FIFO 121, InPort1_FIFO 122 et InPort2_FIFO 123 sont réalisées à partir de la mémoire double port 15 associée au bus *RxBus* 80, un seul registre 21 à trois états est nécessaire entre respectivement la mémoire 15 et le bus *RxBus* 80. Il suffit donc qu'un des signaux *OpenRxTS[0:2]_onRx[0:2]* soit validé pour rendre passant le registre 21 à trois états. Le registre 21 est donc commandé par le signal *OpenRxTS_onRx* 34 généré par le moyen 187 selon la relation suivante :

$$OpenRxTS_onRx = OpenRxTS0_onRx[0:2] \oplus OpenRxTS1_onRx[0:2] \oplus OpenRxTS2_onRx[0:2]$$

De même, un seul registre 14 à trois états entre la mémoire 16 et le bus *TxBus* 70 est commandé par le signal *OpenRxTS_onTx* 33 généré par le moyen 187 selon la relation suivante :

$$OpenRxTS_onTx = OpenRxTS0_onTx[0:2] \oplus OpenRxTS1_onTx[0:2] \oplus OpenRxTS2_onTx[0:2]$$

Le moyen 187 génère aussi des signaux de sortie 31a, 31b, 31c, 81a, 81b et 81c sont connectés au moyen 17 de contrôle de réception RxFIFO.

Les signaux de sortie 37a à 37d et 82a à 82b correspondant respectivement aux 4^{ème}, 5^{ème}, 6^{ème} et 7^{ème} colonnes sont connectés au moyen 130 de contrôle de la FIFO d'entrée. Les signaux de sortie 40a, 40d commandent respectivement les registres trois états 1 à 4 et les signaux de sortie 39a et 39d commandent respectivement les registres à trois état 5 à 8 illustrés en regard de la figure 3.

Pour la signalisation des lignes, des signaux d'entrée incluant principalement des intervalles de temps permettant des cycles 46a, 46b et 46c qui valident les cycles de connexion:

- 5 - pour toute écriture sur les FIFO de sortie respectivement 124, 125 ou 126 à travers les bus *TxBus* 70 ; ou
- pour l'écriture sur les FIFO internes 25, 26 et 27 à travers le bus *RxBus* 80.

On note que durant un même intervalle de temps, une opération d'écriture peut se produire simultanément sur une FIFO de sortie et sur une FIFO interne.

10 Une FIFO interne doit être constamment disponible pour l'écriture afin de respecter une grande vitesse du bus 45 alors que des FIFO de sorties peuvent parfois faire face à des congestions du réseau. C'est la raison pour laquelle les 1^{ère}, 2^{ème} et 3^{ème} lignes comprennent aussi une signalisation complète de FIFO 43a, 43b et 43c des FIFO de sortie respectivement 124, 125 et 126. Tous les signaux
15 d'entrée liés à une ligne sont connectés à chaque point de couplage élémentaire avec cette ligne. Les signaux de sortie résultant pour chaque ligne 42a à 42c et 35a à 35c sont générés par un simple point de couplage élémentaire avec leur ligne correspondante, celle-ci ayant précédemment gagné une connexion avec la ligne.

La figure 5 représente en détail les connexions d'un point de couplage
20 élémentaire 161 représenté en regard de la figure 4, les autres points de couplage élémentaires étant connectés de manière très similaire.

Le point de couplage 161 correspond au moyen d'arbitrage associé à une connexion de la FIFO d'entrée 122 et de la FIFO de sortie 126, tel qu'illustré en regard de la figure 2.

25 Le point de couplage 161 est en charge de la génération des signaux nécessaires pour le point de couplage central 181 illustré en figure 4, en fonction des résultats d'arbitrage.

Pour la gestion des connexions, le point de couplage utilise des signaux 4700b, 303, 304 et 305 en entrée alors qu'il génère des signaux 308a et 309.

Le signal 309 est chaîné en entrée du point de couplage suivant de la même ligne (c'est-à-dire le point 162), alors que le signal 303 provient du point de couplage précédent sur la même ligne (c'est à dire le point 160).

Le signal *Rx1Want2* 4700b est l'un des signaux 47b de la figure 4 indiquant le statut de requête de la connexion entre les FIFO d'entrée 122 et de sortie 126.

Le signal *Prev_wanted[1]* donne le statut courant de la disponibilité d'un intervalle en regard des autres FIFO d'entrée challenger de la même ligne.

Le signal *Fair_arb* 304 est utilisé en option pour donner un niveau de priorité de l'arbitrage durant le chaînage en cascade.

Le signal *EOP_sent* 305 (étant ici *Rx1EOP_sent* en particulier tel que représenté en regard de la figure 5 mais pouvant représenter plus généralement l'un quelconque des signaux *Rx[0:2]_EOP_sent* ou *Tx[0:3]EOP_sent*) et provient d'un point central de couplage 181 et utilisé pour le relâchement d'une connexion dès que la dernière donnée d'un paquet courant a été transmise d'une FIFO d'entrée 122 vers des FIFO sélectionnées de sortie de la 2^{ème} colonne.

Le signal *Connected[2]* 308a informe de manière synchrone le point de couplage central 181 du résultat de l'arbitrage de la connexion courante correspondant au point de couplage 161 alors que le signal *Next_wanted[1]* est cascadié avec le point de couplage 162 pour permettre cet arbitrage.

Pour la gestion du multiplexage par répartition dans le temps impliquant un point de couplage 181, le point de couplage 161 élémentaire génère un signal *RX1Writing_window2_on_Txbus* 310a, alors qu'il reçoit du même point central de couplage un signal *OpenRXTS1_on_Tx2* 301, un signal *EOP_sent* 305, utilisant aussi un intervalle de temps permettant un cycle *EN_cycle2* 46c, et une indication de statut FIFO pleine pour la FIFO de sortie 126 à travers un signal *TxFIFOfull2* 43c.

Le point de couplage 161 est aussi chargé de la génération de l'écriture d'un signal *WriteTxFIFO2* 42c destiné à la FIFO de sortie 126.

La figure 6 représente un point de couplage central 181 de la 2^{ème} colonne tel qu'illustré en regard de la figure 4, les autres points de couplage central étant tout à fait similaire.

Le point de couplage 181 est chargé de la gestion du multiplexage de répartition dans le temps, associé au transfert de tous les paquets de la FIFO d'entrée 121.

Il reçoit d'abord tous les statuts de réservation d'intervalles issus des résultats d'arbitrage des points de couplage élémentaires (151, 156, 161 et 170) en utilisant un signal *Connected[0:3]* 308, et les requêtes associées de transfert des FIFO d'entrées 122 utilisant un signal *Rx1Want[0:3]* 470b qui appartiennent au signal *Rx[0:2]Want[0:3]* 47 généré par le module 20 illustré en regard de la figure 3.

Il reçoit aussi, des mêmes points de couplage central des requêtes d'écriture sur les FIFO de sorties relatives aux réservations d'intervalles :

- *Rx1Writing_window[0:2]_on_Rxbus* 313 ; et
- *Rx1Writing_window[0:2]_on_Txbus* 310.

Le point de couplage central reçoit aussi le signal *Rx1_read* 32b provenant du module 17 de moyen de contrôle de réception de la FIFO illustrée en figure 4, et activé à chaque fois qu'une opération de lecture est effectuée sur une FIFO de sortie associée au port entrant 1.

Le point de couplage central 161 génère un signal *Rx1_connected* 81b vers les moyens 17 de contrôle de FIFO de réception de la figure 3, utilisés pour indiquer si toutes les FIFO de sorties sélectionnées ont été réservées avec succès dans l'étape de réservation d'intervalle.

Lorsque les connexions sont disponibles pour toutes les FIFO de sortie, désirées, le point central de couplage génère les signaux de contrôle *OpenRxTs1_onTx[0:2]* 330b et *OpenRxTs1_onRx[1]* 340b qui sont concaténés par une opération de type « OU » pour générer un signal ouvrant de registre trois états et associés au transfert qui suit, entre une FIFO d'entrée 122 et des FIFO de

sorties sélectionnées (*OpenRxTS1_on_Rx* 34b pour le bus *RxBus* et *OpenRxTS1_on_Tx* 33b pour le bus *TxBus*).

Les signaux *OpenRxTS1_on_Rx* et *OpenRxTS1_on_Tx* issus des points centraux de couplage (180, 181 et 182) 34b et 33b sont respectivement concaténés par une opération de type « OU » dans le but de générer des signaux de contrôle *OpenRxTS_onTx* 33 et *OpenRxTS_onRx* 34 ouvrant respectivement les registres trois états 14 et 21 contrôlant l'accès aux bus *TxBus* et *RxBus* de la figure 3.

Le point central de couplage 181 délivre à chaque point de couplage élémentaire de la même colonne les signaux *OpenRxTS1_onTx[0:2]* 330b, *OpenRxTS1_onRx[1]* 340b. Le point de couplage 161 reçoit alors simplement le signal *OpenRxTS1_onTx[2]* 301 de la figure 5.

Chaque fois qu'une donnée est transmise vers une FIFO de sortie sélectionnée, un point de couplage central 181 active un signal *AllRxData_used1* 31b vers des moyens de contrôle de FIFO de réception, afin de modifier l'adresse de lecture des mémoires 15 et 16 de la figure 3.

Le point de couplage 181 reçoit du module 20 d'analyse d'en-tête de réception de la figure 3 l'indication de la dernière donnée lue, du paquet courant à transmettre vers les FIFO de sortie sélectionnées, en utilisant un signal *RxEOP1* 471b.

Lorsque cette dernière donnée est transmise, le point de couplage 181 active un signal *EOP_sent* vers tous les points de couplage élémentaires de la même colonne dans le but de relâcher la réservation des intervalles de temps correspondant au paquet.

La figure 7 illustre un algorithme d'arbitrage dans un point de couplage élémentaire 161, ce dernier affectant la gestion de la connexion entre une FIFO d'entrée 122 et une FIFO de sortie 126 tel qu'illustré en regard de la figure 2.

Après une étape d'initialisation E0 (« reset »), au cours d'une étape E6 de remise à zéro, les signaux *Connected[2]* et *Next_wanted[1]* sont remis à un niveau logique égal à zéro, indiquant une disponibilité d'intervalle pour une

connexion entre une autre FIFO d'entrée (que la FIFO 122) et la FIFO de sortie 126.

Ensuite, au cours d'étape E7, un test est effectué sur le statut du signal *RxIWant2* qui indique si une requête de connexion est active entre les FIFO 122 et 126. L'étape E7 reste active jusqu'à l'occurrence d'une nouvelle requête (c'est-à-dire jusqu'à l'activation du signal *RxIWant[2]* à un niveau 1).

Après l'occurrence de cette requête, au cours d'une étape E9, un test est effectué sur le signal *Prev_Wanted[1]*.

Si le signal *Prev_Wanted[1]* vaut 1 (signifiant que la connexion est déjà requise par une FIFO d'entrée), au cours d'une étape E50, le signal *Next_Wanted[2]* est mis à un niveau logique égal à 1, propageant ainsi l'état de la connexion jusqu'au point de couplage suivant. Ensuite, l'étape E9 est répétée.

Si au cours l'étape E9, le signal *Prev_Wanted[1]* (qui, pour mémoire, donne le statut courant de la disponibilité d'un intervalle en regard des autres FIFO d'entrée challenger de la même ligne) vaut zéro (signifiant ainsi que la connexion est possible), au cours d'une étape E10, les deux signaux *Connected[2]* et *Next_Wanted[2]* sont mis à un niveau logique égal à 1, et la connexion est ensuite réservée durant tout le transfert du paquet de la FIFO d'entrée 122 vers la FIFO de sortie 126 telles qu'illustrées en regard de la figure 2.

Ensuite au cours d'une étape E11 le moyen 60 d'arbitrage attend que le signal *EOP_sent*, qui correspond à l'indication de fin de transfert de paquet, soit mis à un état égal à 1.

Puis, lorsque le signal *EOP_sent* vaut effectivement 1, l'étape E6 est répétée afin de permettre une relâche de la connexion pour des requêtes ultérieures.

La figure 8 illustre un algorithme de multiplexage dans un point de couplage présent dans la matrice de commutation de la figure 4. Elle illustre notamment le flot de données tel qu'implanté dans le point de couplage élémentaire 161, afin d'effectuer dans un point de couplage central 181 une

jonction du multiplexage par répartition dans le temps en fonction des connexions impliquant la FIFO entrante 122.

Après une étape d'initialisation (étape E0), au cours d'une étape E1, les signaux *Rx1Writing_window2_on_Txbus* et *WriteTxFIFO2* sont remis à un niveau
5 logique zéro.

Ensuite, un test E2 est effectué sur le signal *Connected[2]* qui indique le statut de la connexion.

Si le signal *Connected[2]* vaut zéro (signifiant qu'aucune connexion n'est active), l'étape E1 est réitérée.

10 Si le signal *Connected[2]* vaut 1, une connexion est active. Un test E3 est alors effectué sur les signaux *TxFIFOfull[2]* et *EN_cycle2*.

Le signal *EN_cycle2* oscille de manière permanente à une fréquence égale à la fréquence d'un quart de cycle, et est alloué simultanément aux deux connexions impliquant la FIFO de sortie 126 de la figure 2 et la FIFO de sortie 27
15 O_FIFO_2 de la figure 3, associées chacune de manière indépendante respectivement au bus *TxBus* 70 et au bus *RxBus* 80.

Le signal *TxFIFOfull[2]* indique le statut du remplissage de la FIFO de sortie 126.

Au cours du test E3, si au même moment le signal *TxFIFOfull[2]* n'est pas
20 égal à zéro et si le signal *EN_cycle2* n'est pas égal à 1, l'étape E1 est réitérée.

Dans les autres cas, au cours d'une étape E4,

- le signal *Rx1Writing_window2_on_Txbus* est mis à un niveau logique égal à 1 ; et
- le signal *WriteTxFIFO2*, est mis à niveau égal à celui du signal
25 d'entrée *OpenRXTS1_on_Tx2*, pour un cycle d'horloge.

Ensuite l'étape E1 est réitérée.

La figure 9 illustre un algorithme implanté à l'intérieur du point central de couplage 181 de la 2^{ème} colonne, illustré en regard de la figure 4.

Après une étape d'initialisation (étape E0), au cours d'une étape E12, les
30 signaux suivants sont mis à l'état logique 0 :

- *AllRxdata_used1* ;
- *OpenRx1_onTx[i]* pour une variable *i* allant de 0 à 3 ;
- *OpenRxTs_onRx[1]* ; et
- *Flag EOP*.

5 L'indice *i* indique le nombre possible de connexions entre une FIFO d'entrée 122 et des FIFO de sortie. Ce nombre est incrémenté à chaque première connexion possible de la colonne considérée de la matrice de la figure 4, en commençant par la ligne du haut. Pour la 2^{ème} colonne, une valeur égale à 3 est affectée à la FIFO de sortie 26 O_FIFO_1 de la figure 3.

10 Puis, au cours d'une étape E13, un test est effectué pour vérifier que toutes les demandes de connexion ont abouti (dans ce cas, *Rxwant[0:3]* est égal au signal *Connected[0:3]*).

Si ce n'est pas le cas, l'étape E12 est réitérée

Si toutes les demandes de connexion ont abouti, au cours d'une étape E14,
15 le signal *AllRxdata_used1* est mis à niveau logique 1 durant un cycle d'horloge, afin de valider la lecture d'une donnée élémentaire dans les mémoires 15 et 16 de la figure 4 associées à la FIFO d'entrée 122.

Puis, au cours d'une étape E15, on attend que le signal *Rx1_read* vaille 1.

Lorsque le signal *Rx1_read* vaut 1, on commence une boucle commençant
20 à l'étape E16.

Cette boucle permet de s'assurer que pour les connexions établies, chaque donnée lue à partir de la FIFO d'entrée 122 est transférée vers les FIFO de sortie sélectionnées.

Au cours du test E16, le moyen d'arbitrage 60 détermine si le signal
25 *Connected[i]* vaut 1.

Dans la négative, le signal *Connected[i]* de 1 vaut zéro, correspondant ainsi à une connexion absente, au cours d'une étape E21, la valeur de l'indice *i* est comparée à 3.

Si l'indice *i* n'est pas égal à 3, au cours d'une étape E20, l'indice *i* est
30 incrémenté d'une valeur égale à 1 et l'étape E16 est réitérée.

Si au cours de l'étape E21, l'indice i vaut 3, au cours d'une étape E24 le signal *AllRxData_used1* est mis à 1 alors que le signal *EOP_sent* prend la valeur du signal *Flag_eop* durant un cycle d'horloge.

Puis au cours d'une étape E23, un test effectué sur les signaux internes
 5 *Flag_eop*.

Le signal *Flag_eop* utilisé pour permettre la transmission de la dernière donnée d'un paquet courant sur toutes les connexions actives alors qu'une indication de dernière donnée lue est générée par le module 19 illustré en regard de la figure 3.

10 Si au cours de l'étape E23, le signal *Flag_eop* vaut 1, les connexions sont relâchées et l'étape E12 est réitérée.

Si au cours de l'étape E23, le signal *Flag_eop* vaut zéro, au cours d'une étape E25 un test est effectué sur le signal *Rx1_read*, correspondant à la disponibilité d'une nouvelle donnée à transférer vers les FIFO de sortie.

15 Tant que le signal *Rx1_read* n'est pas égal à 1, l'opération E25 est réitérée.

Lorsque le signal *Rx1_read* vaut 1, au cours de l'étape E26 un test est effectué sur le signal *Rx1_eop*.

Si le signal *Rx1_eop* vaut 1, au cours d'une étape E27, le signal *Flag_eop* est mis à l'état logique 1 (pour permettre la mémorisation des indications de la
 20 dernière donnée à transmettre aux FIFO de sortie). Puis, l'étape E16 est réitérée.

Lors de l'étape E26, si le signal *Rx1_eop* vaut zéro, au cours d'une étape E19, le signal *Flag_eop* est remis à l'état logique zéro. Puis l'étape E16 est réitérée.

Si au cours de l'étape E16, le signal *connected[i]* vaut zéro, au cours d'une
 25 étape E17, un test est effectué en fonction de la valeur de l'indice i :

si l'indice i vaut 3, le test E17 est effectué sur le signal *Rx1Writing_window[1]_on_Rxbus* (correspondant à la fenêtre de triggering pour le transfert des données vers la FIFO de sortie O_FIFO_1 26)

- si l'indice i n'est pas égal à 3, le test effectué sur le signal $RxWriting_window[i]_{on_Txbus}$ (correspondant à la valeur des triggers pour le transfert de données vers les FIFO de sortie 124, 125 ou 126 en fonction de la valeur de l'indice i).

5 Dans tous les cas, l'étape E17 est effectuée tant que les signaux testés relatifs ne valent pas 1.

Lorsque le signal testé vaut 1, une étape E18 est effectuée en fonction de la valeur de l'indice i :

- si l'indice i vaut 3, le signal $OpenRxTs1_{on_Rx}[1]$ est mis à un niveau
10 logique égal à 1 durant un cycle d'horloge ;
- sinon, le signal $Open_TxTs1_{on_Tx}[i]$ est mis à l'état logique égal à 1 durant un cycle d'horloge.

Ensuite, l'étape E21 est effectuée.

15 En parallèle, le signal de sortie $OpenRxTs1_{on_Rx}$ est mis à une valeur égale à celle du signal $OpenRxTs1_{on_Rx1}$.

En parallèle, les signaux $Open_RxTs1_{on_Tx1}$, et $Open_RxTs1_{on_Tx2}$ sont additionnés (fonction « ou » en logique booléenne) pour générer un signal de sortie $OpenRxTs1_{on_Tx}$.

20 Ces signaux de sortie sont à nouveau sommés (non illustré) dans le module 60 d'arbitrage, illustré en regard de la figure 3, pour générer respectivement des signaux $OpenRxTs_{on_Rx}$ et $Open_RxTs_{on_Tx}$.

25 La figure 10 décrit un diagramme temporel de la gestion du MRT effectué par le point de couplage central 181 de la figure 4 lorsqu'un paquet est transféré d'une FIFO d'entrée 122 vers des FIFO de sorties 124 et 126. Le paquet est étiqueté avec une valeur n et sa longueur est égale à m , correspondant au nombre de données élémentaires.

Les cycles d'horloges sont indiqués avec des références p à $p+17$ et k à $k+14$, le passage entre $p+17$ et k représenté par le signe « // » symbolisant une durée plus ou moins longue n'étant pas détaillé.

Le diagramme de temps montre les trois phases d'un processus de communication correspondant à un cycle de connexion :

- une phase de connexion ;
- une phase de transfert ; et
- 5 - une phase de déconnexion.

La phase de connexion se termine au cycle $p+4$ lorsque le signal *Rx1_connected* devient actif, indiquant ainsi le succès d'une réservation d'intervalle pour les FIFO de sortie 124 et 126.

10 Sur la figure 10, les signaux de requête et de chaînage ne sont pas représentés.

La phase de transfert commence dès que la phase de connexion est terminée. Les fenêtres de « triggering » (ou cycle d'écriture possible vers une FIFO associée à un port de sortie) sont alors validées (*Rx1Writing_window0_on_Txbus* pour la FIFO de sortie 124 et
15 *Rx1Writing_window2_on_Txbus* pour la FIFO de sortie 126). On note qu'un cycle d'écriture possible dans une FIFO est notamment conditionné par l'état de cette FIFO (par exemple, le signal *TxFIFOfull[0:2]* indiquant que la ou les FIFOs pleines)

Le signal *AllRxData_used1* est actif durant le cycle $p+6$ pour forcer la
20 lecture d'une première donnée $Dn,1$ sur la mémoire relative à la FIFO d'entrée 122 par activation du signal *Rx1_read* durant le cycle $p+7$.

Lorsque la première donnée $Dn,1$ est lue, le signal *OpenRxTs_on_Tx* prend les mêmes valeurs qu'une fenêtre de triggering durant toute la phase de transfert de la première donnée, et est activé en parallèle avec les signaux d'écriture sur les
25 FIFO de sortie (signaux *WriteTxFIFO0* et *WriteTxFIFO2*).

Ainsi, lorsque le signal *Rx1Writing_window0_on_Txbus* est activé au cours du cycle $p+9$, une écriture de la première donnée $Dn,1$ a lieu sur la FIFO de sortie 124. De même, lorsque le signal *Rx1Writing_window2_on_Txbus* est activé au cours du cycle $p+11$, une écriture de la première donnée $Dn,1$ a lieu sur la



FIFO de sortie 122. On note que ces opérations d'écritures de la donnée $Dn,1$ sont effectuées au cours d'une phase de 4 cycles élémentaires référencés $p+9$ à $p+12$.

Lorsqu'une écriture a eu lieu sur chacune des FIFO de sortie (ici au cours du cycle $p+11$), le signal *AllData_used1* est activé durant un cycle ($p+11$),
 5 attendant ainsi l'occurrence de validité du signal *Rx1_read* au cours du cycle $p+12$ permettant le transfert de nouvelles données.

Du cycle $p+11$ jusqu'au cycle $p+14$, la FIFO 124 est pleine (le signal *TXFIFOfull0* vaut alors 1), invalidant alors la fenêtre de triggering relative à la FIFO de sortie 124 durant au moins quatre cycles.

10 Au cours du cycle $p+12$ (signal *Rx1_read* activé), une nouvelle donnée $Dn,2$ est lue de la mémoire de type FIFO, mais n'est pas transmise en attendant la disponibilité d'un nouvel intervalle de temps pour une connexion suivante (ici l'occurrence suivante de la fenêtre de triggering indiqué par la validité du signal *Rx1Writing_window0_on_Txbus*).

15 L'interruption de transfert prend fin au cycle $p+15$.

Sur le cycle $k+6$, la dernière donnée du paquet Dn,m est lue de la FIFO d'entrée 122. La phase de transfert se termine alors lorsque cette donnée Dn,m est transmise vers les FIFO de sorties 124 et 126 après un cycle $k+9$, alors que le signal *EOP_sent* est actif.

20 La phase de déconnexion est immédiatement effectuée sur les points de couplage impliqués (151,161 et 181), et le signal *Rx1_connected* est relâché pour des connexions futures.

Bien entendu, l'invention n'est pas limitée aux exemples de réalisation mentionnés ci-dessus.

25 En particulier, l'homme du métier pourra apporter toute variante dans la définition de l'architecture du dispositif de commutation qui pourra notamment comprendre plusieurs interfaces de type IEEE1394 ou IEEE1355 ou qui ne comprendra pas l'un ou l'autre des interfaces de type IEEE1394 ou IEEE1355.

30 On note également que l'invention ne se limite pas aux dispositifs de commutation en tant que tel mais s'étend à tout appareil assurant une fonction de

commutation adaptée à effectuer des opérations de transmission de paquets point vers multipoint au cours d'un cycle de connexion.

On note aussi que l'invention s'applique aux domaines suivants:

- la commutation à haut débit ;
- 5 - les applications distribuées ;
- la transmission de données numériques ;
- la réception de données numériques ;
- les applications audio ;
- les réseaux d'entreprise ; et
- 10 - la transmission d'images en temps réel.

On notera que l'invention ne se limite pas à une implantation purement matérielle mais qu'elle peut aussi être mise en œuvre sous la forme d'une séquence d'instructions d'un programme informatique ou toute forme mixant une partie matérielle et une partie logicielle. Dans le cas où l'invention est implantée

15 partiellement ou totalement sous forme logicielle, la séquence d'instructions correspondante pourra être stockée dans un moyen de stockage amovible (tel que par exemple une disquette, un CD-ROM ou un DVD-ROM) ou non, ce moyen de stockage étant lisible partiellement ou totalement par un ordinateur ou un microprocesseur.

20

REVENDICATIONS

1. Procédé de commutation d'unités de données de protocole mettant en œuvre des moyens de commutation permettant d'interconnecter sélectivement un port émetteur et au moins un port récepteur sélectionné parmi au moins deux ports récepteurs, par l'intermédiaire d'au moins un bus interne (*RxBus* 80, *TxBus* 70), chacune des unités de données de protocole étant constituée d'au moins une donnée élémentaire,
- 5 caractérisé en ce qu'il met en œuvre :
- 10 - un mécanisme de synchronisation définissant des intervalles temporels, dits cycles de connexion, sur au moins un desdits bus internes (*RxBus* 80, *TxBus* 70) ;
- un mécanisme d'allocation d'au moins un desdits cycles de connexion à chacun desdits ports récepteurs sélectionnés; et
- 15 - un mécanisme d'écriture d'au moins une donnée élémentaire dans le ou les cycles de connexion alloués, de façon à permettre la diffusion desdites données élémentaires auxdits ports récepteurs sélectionnés.
2. Procédé selon la revendication 1 caractérisé en ce que ledit mécanisme d'écriture comprend une étape de vérification déterminant si chacune desdites données élémentaires a été reçue par chacun desdits ports récepteurs sélectionnés.
- 20 3. Procédé selon la revendication 2 caractérisé en ce que ledit mécanisme d'écriture de donnée élémentaire dans chacun desdits cycles de connexion alloués est réitéré tant que ladite donnée élémentaire n'a pas été reçue par tous les ports récepteurs.
- 25 4. Procédé selon l'une quelconque des revendications 1 à 3 caractérisé en ce que ledit mécanisme d'allocation comprend une étape d'association de chacun desdits cycles de connexion à chacun desdits ports récepteurs sélectionnés.
5. Procédé selon l'une quelconque des revendications 1 à 4, caractérisé en ce que ledit mécanisme d'allocation comprend :

- une étape de détection (E7) d'un port émetteur requérant le transfert d'au moins une unité de données de protocole vers au moins un port récepteur sélectionné;
 - une étape de vérification (E9) que le ou lesdits ports récepteurs
5 sélectionnés sont prêts à recevoir ladite ou lesdites unités de données de protocole ; et
 - une étape de validation (E10) d'au moins un cycle de connexion permettant l'écriture des données élémentaires de la ou desdites unités de données de protocole dans lesdits ports récepteurs sélectionné durant le ou lesdits cycles de
10 connexion validés, lorsque la vérification est positive.
6. Procédé selon l'une quelconque des revendications 1 à 5, caractérisé en ce qu'il met en œuvre au moins un premier bus d'entrée (*RxBus 80*) multiplexant lesdites données élémentaires provenant d'au moins deux ports émetteurs et/ou au moins un premier bus de sortie (*TxBus 70*) multiplexant lesdites données
15 élémentaires à destination d'au moins deux ports récepteurs.
7. Procédé selon l'une quelconque des revendications 1 à 6, caractérisé en ce que lesdits ports émetteurs (*InPort0*, *InPort1*, *InPort2*) et récepteurs (*OutPort0*, *OutPort1*, *OutPort2*) sont organisés par paires ((*InPort0*, *OutPort0*), (*InPort1*, *OutPort1*), (*InPort2*, *OutPort2*))regroupant chacune un port émetteur et
20 un port récepteur, chaque paire étant associée à un lien distinct.
8. Procédé selon l'une quelconque des revendications 1 à 7, caractérisé en ce qu'il met en œuvre au moins un lien (104, 103, 101) connecté à des ports émetteurs et/ou à des ports récepteurs.
9. Procédé selon la revendication 8, caractérisé en ce que ledit lien appartient
25 au groupe comprenant :
- les liens IEEE 1355 (104) ou équivalent; et
 - les bus externes (101, 103).
10. Procédé selon l'une quelconque des revendications 1 à 9, caractérisé en ce que ledit procédé est apte à commuter des données dont le débit est supérieur ou
30 égal à 100 Mbit/s.

11. Procédé selon l'une quelconque des revendications 1 à 10 caractérisé en ce que ledit mécanisme d'écriture comprend au moins une étape d'écriture de chaque donnée élémentaire de chacune desdites unités de données de protocole, chacune desdites étapes d'écriture de ladite chaque donnée élémentaire comprenant :
- 5 - une sous-étape d'acceptation (E3) par chacun desdits ports récepteurs sélectionné de l'écriture de ladite chaque donnée élémentaire à transmettre ; et
- une sous-étape de transmission (E14) de ladite chaque donnée élémentaire à transmettre, vers chacun desdits ports récepteurs sélectionné.
12. Procédé selon la revendication 11 caractérisé en ce que lors de ladite étape
- 10 d'acceptation (E3), l'acceptation est conditionnée par un degré de remplissage d'une mémoire de réception (124, 125, 126, 25, 26, 27) associée audit port récepteur sélectionné, pour chacun desdits ports récepteurs.
13. Procédé selon l'une quelconque des revendications 1 à 12 caractérisé en ce que ledit mécanisme d'écriture comprend au moins une étape d'arbitrage d'au
- 15 moins un bus reliant un ensemble d'au moins un port d'entrée comprenant ledit port émetteur à un ensemble d'au moins un port de sortie comprenant lesdits ports récepteurs, ladite étape d'arbitrage étant assurée par une matrice de commutation (60) constituée de points de couplage (150 à 186) apte à transmettre des données élémentaires entre un port d'entrée et un port de sortie, et organisée en lignes et en
- 20 colonnes,
- chaque dite colonne (respectivement ligne) étant apte à gérer la réception de données élémentaires en provenance d'un port d'entrée associé à ladite colonne (respectivement ligne); et
- chaque dite ligne (respectivement colonne) étant apte à gérer l'émission de
- 25 données élémentaires à destination d'un port de sortie associé à ladite ligne respectivement colonne);
- de sorte qu'un seul point de couplage par ligne (respectivement colonne) puisse, à un instant donné, permettre la transmission de données élémentaires.

14. Procédé selon l'une quelconque des revendications 1 à 13 caractérisé en ce que chaque dite unité de données de protocole transmise comprend au moins un en-tête et en ce que ledit procédé comprend en outre :
- au moins une étape d'analyse dudit en-tête ; et/ou
 - 5 - au moins une étape de modification dudit en-tête.
15. Dispositif de commutation d'unités de données de protocole mettant en œuvre des moyens de commutation permettant d'interconnecter sélectivement un port émetteur et au moins un port récepteur sélectionné parmi au moins deux ports récepteurs, par l'intermédiaire d'au moins un bus interne (*RxBus* 80, *TxBus* 70),
- 10 chacune des unités de données de protocole étant constituée d'au moins une donnée élémentaire,
- caractérisé en ce qu'il met en œuvre :
- un moyen de synchronisation définissant des intervalles temporels, dits cycles de connexion, sur au moins un desdits bus internes (*RxBus* 80, *TxBus* 70) ;
 - 15 - un moyen d'allocation d'au moins un desdits cycles de connexion à chacun desdits ports récepteurs sélectionnés ; et
 - un moyen d'écriture d'au moins une donnée élémentaire dans le ou les cycles de connexion alloués, de façon à permettre la diffusion desdites données élémentaires auxdits ports récepteurs sélectionnés.
- 20 16. Dispositif selon la revendication 15 caractérisé en ce que ledit moyen d'écriture comprend un moyen de vérification déterminant si chacune desdites données élémentaires a été reçue par chacun desdits ports récepteurs sélectionnés.
17. Dispositif selon la revendication 16 caractérisé en ce que ledit moyen d'écriture de donnée élémentaire dans chacun desdits cycles de connexion alloués
- 25 est mis en œuvre tant que ladite donnée élémentaire n'a pas été reçue par tous les ports récepteurs.
18. Dispositif selon l'une quelconque des revendications 15 à 17 caractérisé en ce que ledit moyen d'allocation comprend une étape d'association de chacun desdits cycles de connexion à chacun desdits ports récepteurs sélectionnés.

19. Dispositif selon l'une quelconque des revendications 15 à 18, caractérisé en ce que ledit moyen d'allocation comprend :
- un moyen de détection (E7) d'un port émetteur requérant le transfert d'au moins une unité de données de protocole vers au moins un port récepteur sélectionné;
 - un moyen de vérification (E9) que le ou lesdits ports récepteurs sélectionnés sont prêts à recevoir ladite ou lesdites unités de données de protocole ; et
 - un moyen de validation (E10) d'au moins un cycle de connexion permettant l'écriture des données élémentaires de la ou desdites unités de données de protocole dans lesdits ports récepteurs sélectionné durant le ou lesdits cycles de connexion validés, lorsque la vérification est positive.
20. Dispositif selon l'une quelconque des revendications 15 à 19, caractérisé en ce qu'il met en œuvre au moins un premier bus d'entrée (*RxBus* 80) multiplexant lesdites données élémentaires provenant d'au moins deux ports émetteurs et/ou au moins un premier bus de sortie (*TxBus* 70) multiplexant lesdites données élémentaires à destination d'au moins deux ports récepteurs.
21. Dispositif selon l'une quelconque des revendications 15 à 20, caractérisé en ce que lesdits ports émetteurs (*InPort0*, *InPort1*, *InPort2*) et récepteurs (*OutPort0*, *OutPort1*, *Outport2*) sont organisés par paires ((*InPort0*,*OutPort0*), (*InPort1*,*OutPort1*), (*InPort2*,*OutPort2*))regroupant chacune un port émetteur et un port récepteur, chaque paire étant associée à un lien distinct.
22. Dispositif selon l'une quelconque des revendications 15 à 21, caractérisé en ce qu'il met en œuvre au moins un lien (104, 103, 101) connecté à des ports émetteurs et/ou à des ports récepteurs.
23. Dispositif selon la revendication 22, caractérisé en ce que ledit lien appartient au groupe comprenant :
- les liens IEEE 1355 (104) ou équivalent; et
 - les bus externes (101, 103).

24. Dispositif selon l'une quelconque des revendications 15 à 23, caractérisé en ce que ledit dispositif est apte à commuter des données dont le débit est supérieur ou égal à 100 Mbit/s.

5 25. Dispositif selon l'une quelconque des revendications 15 à 24 caractérisé en ce que ledit moyen d'écriture met en oeuvre une étape d'écriture de chaque donnée élémentaire de chacune desdites unités de données de protocole, et comprend lui-même:

- un moyen d'acceptation (E3) par chacun desdits ports récepteurs sélectionné de l'écriture de ladite chaque donnée élémentaire à transmettre ; et
- 10 - un moyen de transmission (E14) de ladite chaque donnée élémentaire à transmettre, vers chacun desdits ports récepteurs sélectionné.

26. Dispositif selon la revendication 25 caractérisé en ce que l'acceptation mise en oeuvre par ledit moyen d'acceptation (E3) est conditionnée par un degré de remplissage d'une mémoire de réception (124, 125, 126, 25, 26, 27) associée
15 audit port récepteur sélectionné, pour chacun desdits ports récepteurs.

27. Dispositif selon la revendication 26 caractérisé en ce que ladite mémoire comprend au moins une FIFO.

28. Dispositif selon l'une quelconque des revendications 15 à 27 caractérisé en ce que ledit moyen d'écriture comprend au moins un moyen d'arbitrage d'au
20 moins un bus reliant un ensemble d'au moins un port d'entrée comprenant ledit port émetteur à un ensemble d'au moins un port de sortie comprenant lesdits ports récepteurs, ledit moyen d'arbitrage mettant en oeuvre une matrice de commutation (60) constituée de points de couplage (150 à 186) apte à transmettre des données élémentaires entre un port d'entrée et un port de sortie, et organisée en lignes et en
25 colonnes,

- chaque dite colonne (respectivement ligne) étant apte à gérer la réception de données élémentaires en provenance d'un port d'entrée associé à ladite colonne (respectivement ligne); et

- chaque dite ligne (respectivement colonne) étant apte à gérer l'émission de données élémentaires à destination d'un port de sortie associé à ladite ligne respectivement colonne);

de sorte qu'un seul point de couplage par ligne (respectivement colonne) puisse, à un instant donné, permettre la transmission de données élémentaires.

29. Dispositif selon l'une quelconque des revendications 15 à 28 caractérisé en ce que chaque dite unité de données de protocole transmise comprend au moins un en-tête et en ce que ledit dispositif comprend en outre :

- au moins un moyen d'analyse dudit en-tête ; et/ou
- au moins un moyen de modification dudit en-tête.

30. Dispositif selon l'une quelconque des revendications 15 à 29 caractérisé en ce qu'il comprend en outre un moyen d'interfaçage (104), délivrant à un module de contrôle (107) et via un moyen de transmission de signaux d'horloges, des signaux d'horloge régénérés à partir de paquets reçus par ledit moyen d'interfaçage.

31. Dispositif selon l'une quelconque des revendications 15 à 29 caractérisé en ce qu'il comprend en outre un moyen d'interfaçage (104) transmettant et/ou recevant des informations via au plus deux bus de liaison (*RxDI 44*, *TxDO 41*) à destination et/ou en provenance d'au moins un des moyens appartenant au groupe comprenant ledit moyen de synchronisation, ledit moyen d'allocation et ledit moyen d'écriture.

32. Dispositif selon la revendication 31 caractérisé en ce que les unités de données de protocole émises par au moins un port émetteur (*InPort0*, *InPort1*, *InPort2*) vers des FIFOs (121,122, 123) sont multiplexées sur un bus de liaison en réception (*RxDI 44*).

33. Dispositif selon l'une quelconque des revendications 31 et 32 caractérisé en ce que les unités de données de protocole reçues par au moins un port récepteur (*OutPort0*, *OutPort1*, *Outport2*) par l'intermédiaire de FIFOs (124,125, 126) sont démultiplexées sur un bus de liaison en émission (*TxDO 41*).

34. Appareil de commutation caractérisé en ce qu'il comprend :

- au moins un dispositif de commutation (105) selon l'une quelconque des revendications 15 à 33 ;
- et au moins un élément appartenant au groupe comprenant
 - les liens IEEE 1355 (104) ou équivalents; et
 - 5 - les bus externes (101, 103).

35. Appareil de commutation selon la revendication 34 caractérisé en ce qu'il est relié à un appareil de traitement de données.

36. Application du procédé selon l'une quelconque des revendications 1 à 14 à l'un au moins des domaines appartenant au groupe comprenant :

- 10 - la commutation à haut débit ;
- les applications distribuées ;
- la transmission de données numériques ;
- la réception de données numériques ;
- les applications audio ;
- 15 - les réseaux d'entreprise ; et
- la transmission d'images en temps réel.

37. Produit programme d'ordinateur caractérisé en ce que ledit programme comprend des séquences d'instructions adaptées à la mise en œuvre d'un procédé selon l'une quelconque des revendications 1 à 14 lorsque ledit programme est
20 exécuté sur un ordinateur.

38. Produit programme d'ordinateur de commutation d'unités de données de protocole mettant en œuvre des moyens de commutation permettant d'interconnecter sélectivement un port émetteur et au moins un port récepteur sélectionné parmi au moins deux ports récepteurs, par l'intermédiaire d'au moins
25 un bus interne, chacune des unités de données de protocole étant constituée d'au moins une donnée élémentaire, ledit produit programme d'ordinateur comprenant des instructions de code de programme enregistré sur un support utilisable dans un ordinateur comprenant :

- des moyens de programmation lisibles par ordinateur pour effectuer une étape de synchronisation définissant des intervalles temporels, dits cycles de connexion, sur au moins un desdits bus internes ;
 - des moyens de programmation lisibles par ordinateur pour effectuer une étape
- 5 d'allocation d'au moins un desdits cycles de connexion à chacun desdits ports récepteurs sélectionnés et
- des moyens de programmation lisibles par ordinateur pour effectuer une étape d'écriture d'au moins une donnée élémentaire dans le ou les cycles de connexion alloués, de façon à permettre la diffusion desdites données
- 10 élémentaires auxdits ports récepteurs sélectionnés.

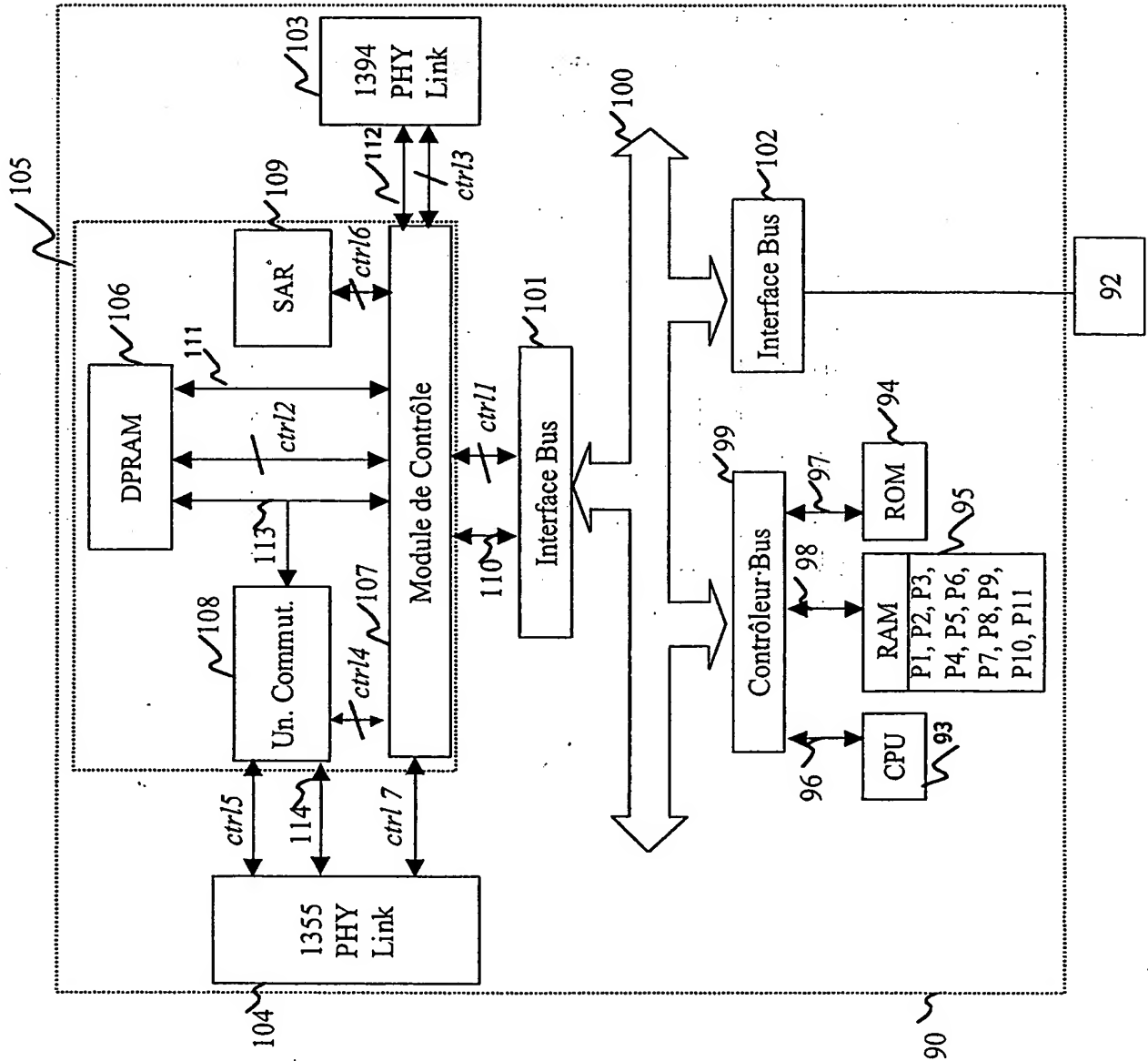


Fig. 1

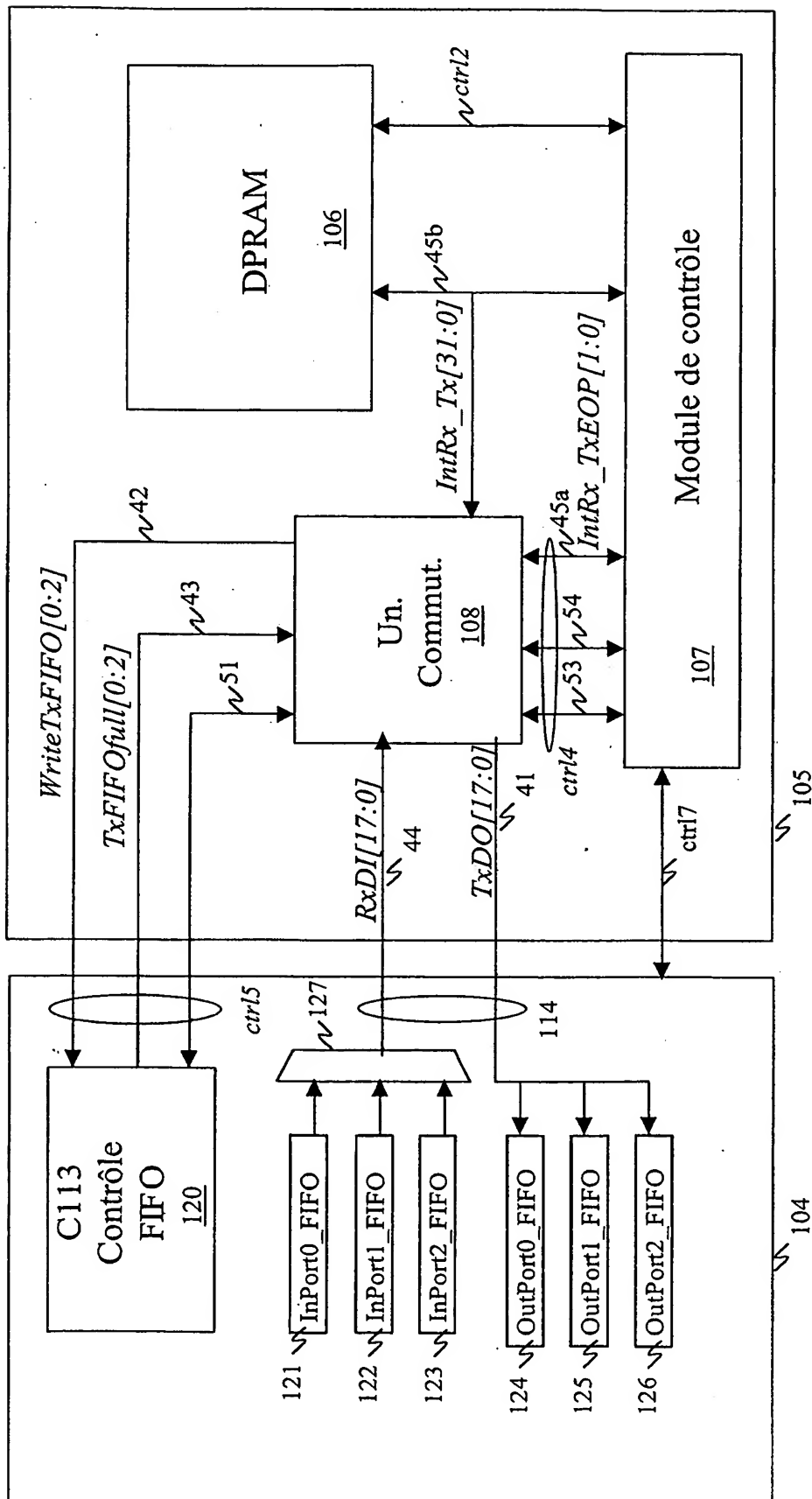


Fig. 2

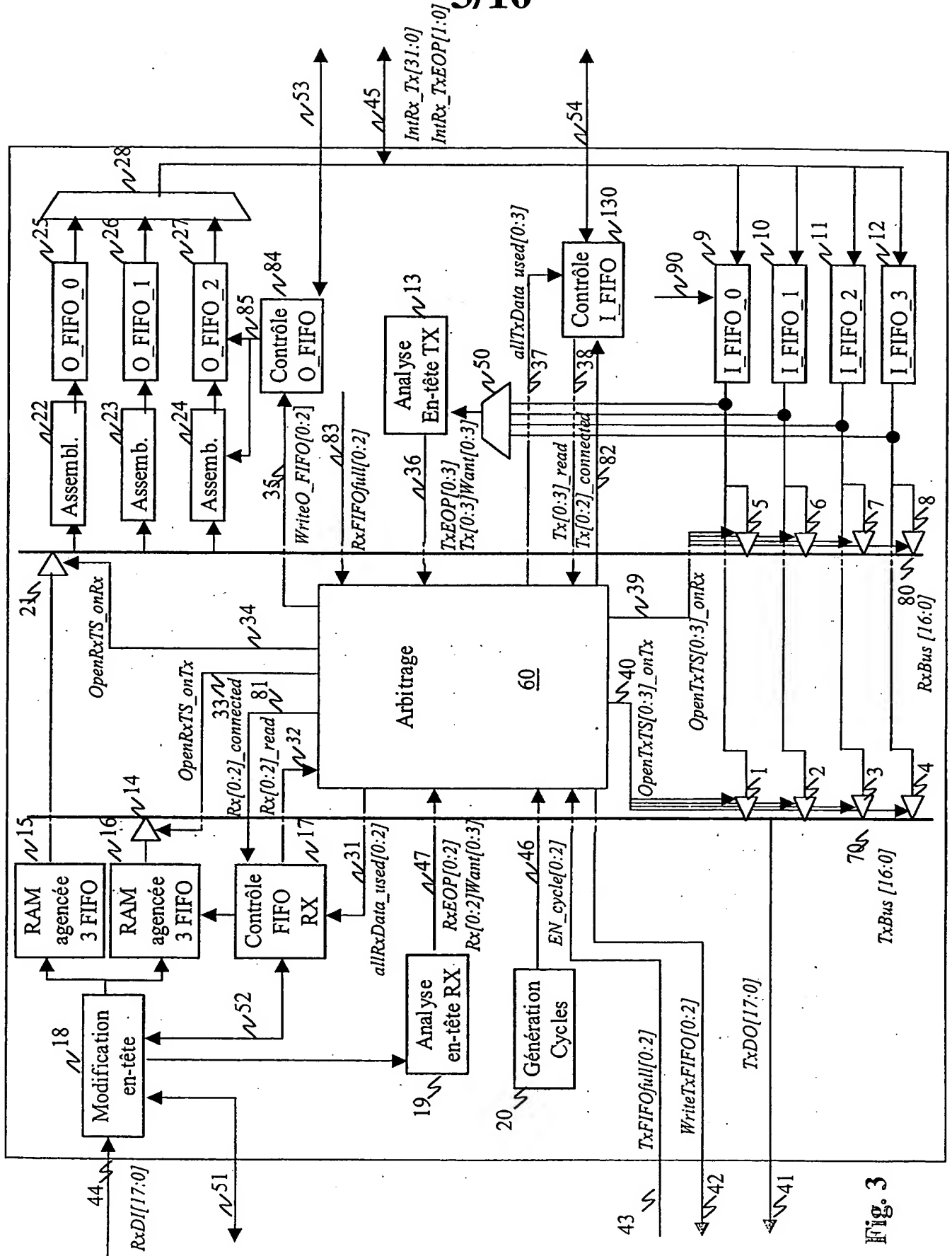


Fig. 3



4/10

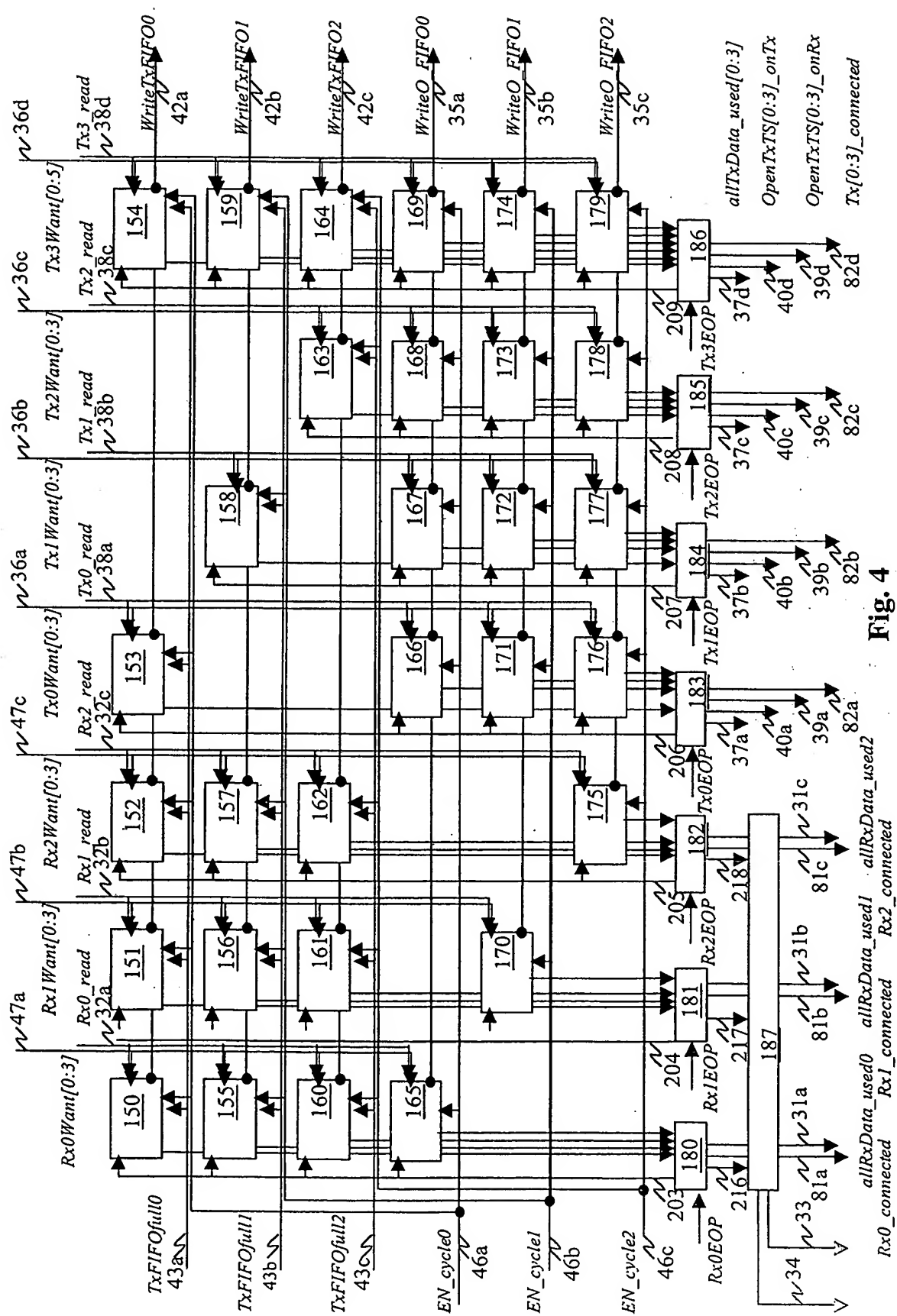


Fig. 4

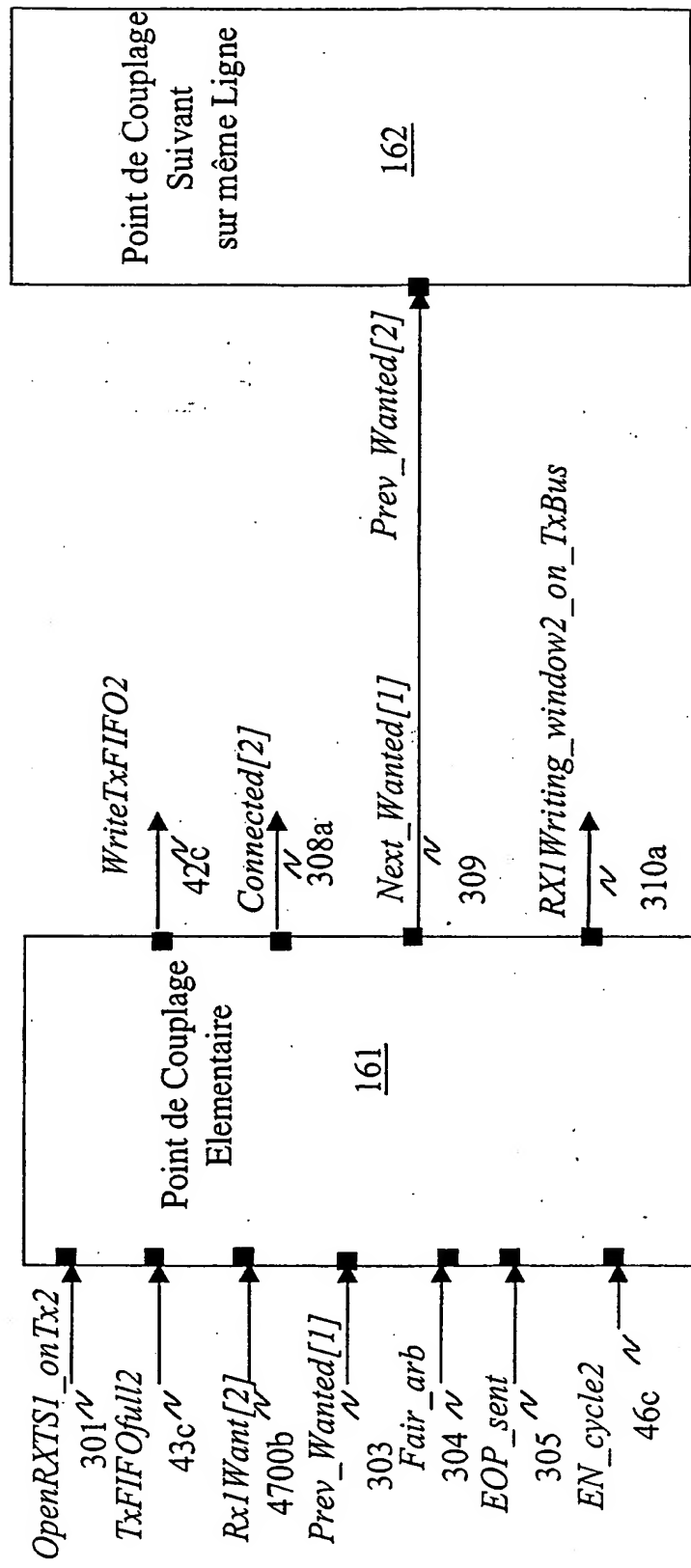


Fig. 5

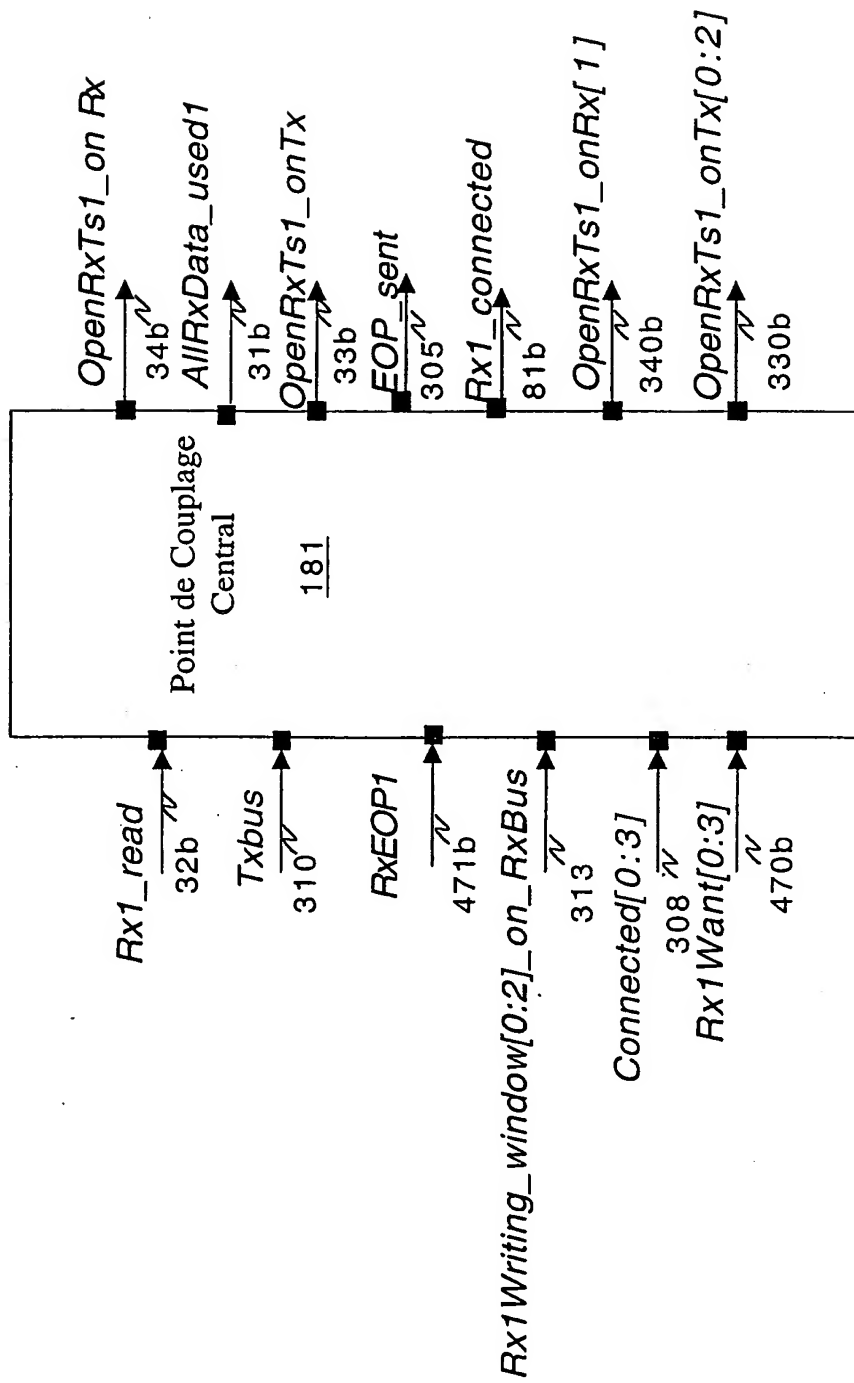


Fig. 6

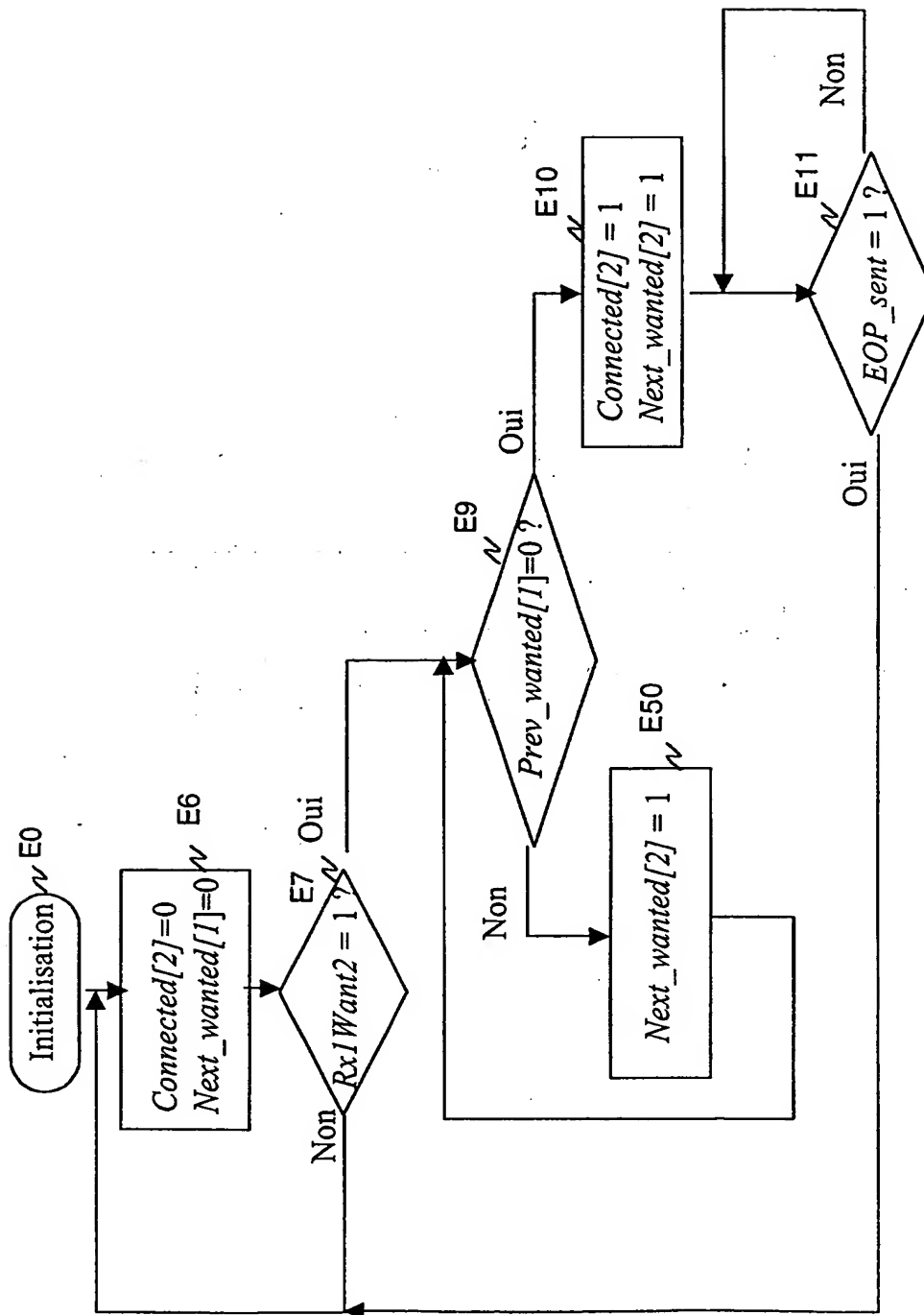
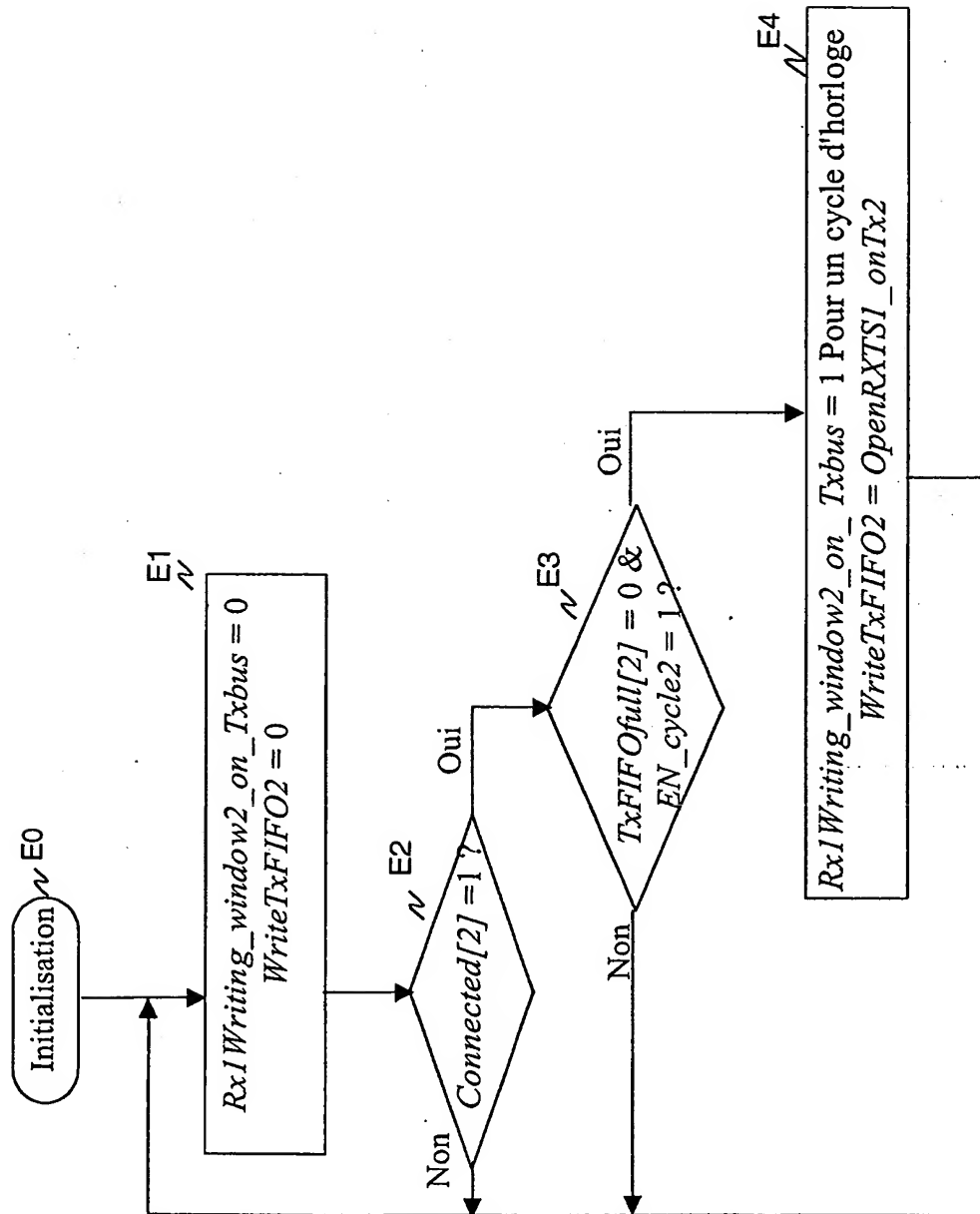


Fig.7



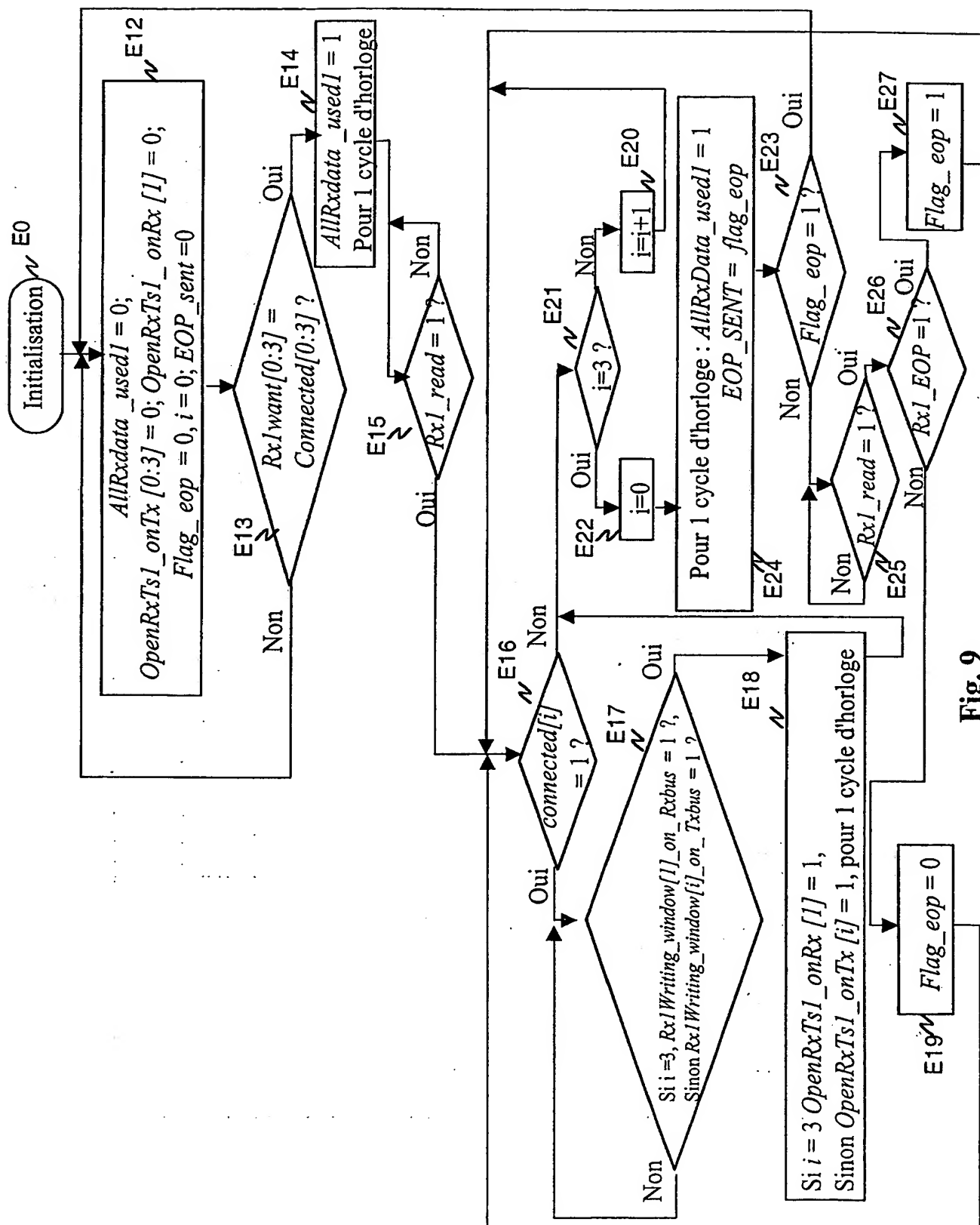
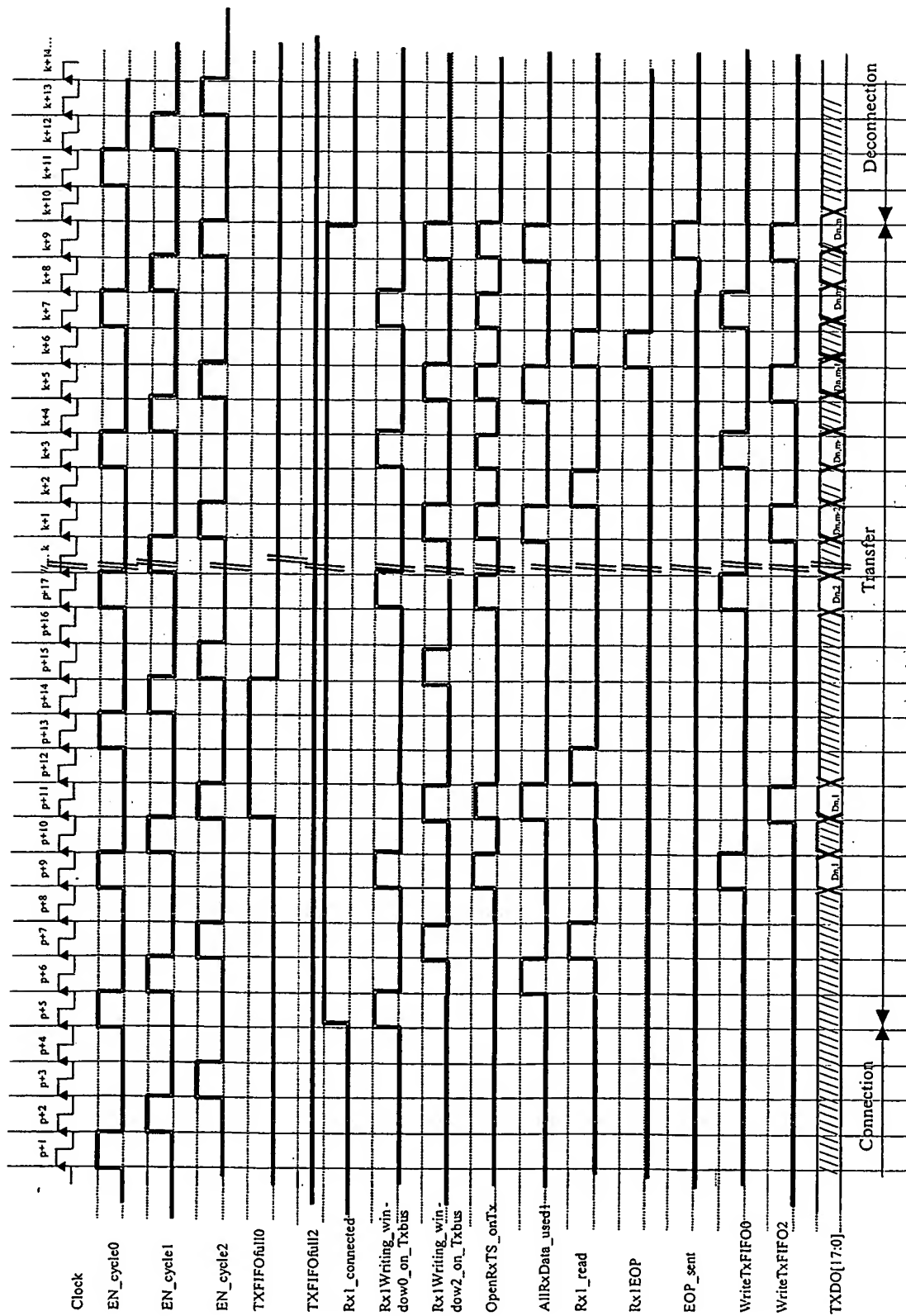


Fig. 9



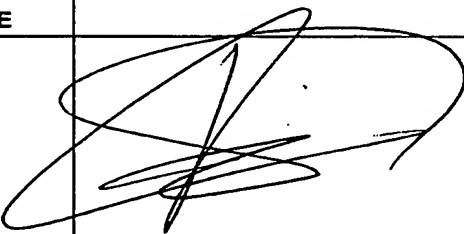


BREVET D'INVENTION

Désignation de l'inventeur

Vos références pour ce dossier	6751
N°D'ENREGISTREMENT NATIONAL	04 02 037
TITRE DE L'INVENTION	
	Dispositif et procédé de transmission dans un commutateur
LE(S) DEMANDEUR(S) OU LE(S) MANDATAIRE(S):	

DESIGNE(NT) EN TANT QU'INVENTEUR(S):	
Inventeur 1	
Nom	FROUIN
Prénoms	Laurent
Rue	78, rue de Sarzeau
Code postal et ville	35700 RENNES
Société d'appartenance	CANON CRF SA
Inventeur 2	
Nom	CLOSSET
Prénoms	Arnaud
Rue	La Fosse au Loup
Code postal et ville	35510 CESSON SEVIGNE
Société d'appartenance	CANON CRF SA

DATE ET SIGNATURE(S) DU (DES) DEMANDEUR(S) OU DU MANDATAIRE	
Signé par:	
Date	14 fév. 2001

La loi n°78-17 du 6 janvier 1978 relative à l'informatique aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, NY 10112-3801

Applicant: Laurent FROUIN
Application No.: 10/055,939
Group Art Unit: Unassigned
Filed: January 28, 2002
Title: DEVICE AND METHOD FOR
TRANSMISSION IN SWITCH

RECEIVED
MAY 14 2002
OIPF/JCW/S